maszyny matematyczne

POLITECHNIKA ŚLĄSKA BIBLIOTEKA GŁÓWNA w Gliwicach
P. 1877/70

zastosowania

w gospodarce

technice

i nauce

11
1970

SPIS TREŚCI

Str.

Jerzy Leszczyński — „System eksploatacji EMC ODRA 1204/N" 1

Bolesław Gliksman — „Struktura, działanie, oprogramowanie elektronicznej maszyny cyfrowej MIŃSK-32" — Cz. II — Software 5

Czesław Paczuła — „Niektóre problemy zastosowania elektronicznej techniki obliczeniowej w budownictwie" 8

Tadeusz Olszewski — „FORPIT — polska modyfikacja 160 FORTRAN-A dla zastosowań metody MONTE-CARLO" 12

Zdenek Křečan — „Maszyny uczące w Czechosłowacji" 16

ZE ŚWIATA

„II Międzynarodowa Konferencja Sztucznej Inteligencji" — A.B.E. . . . 19

„Problemy szkolenia programistów w USA" — **Stefan Kwiatek** 20

„Stany Zjednoczone AP 1970" — K.P. 22

„Konferencje. Wystawy. Targi" — H.Kl. 23

„XVIII Międzynarodowy Kongres Naukowy Elektroniki" . . . . IV str. okł.

PRETO informuje 23

WIADOMOŚCI PKAPI . . . . . III str. okł.

„Seminaria Klubu Użytkowników Komputerów rodziny ICL-1900" — A.B.E.

WYDAWNICTWA CZASOPISM TECHNICZNYCH NOT
Warszawa Czackiego 3/5

KOLEGIUM REDAKCYJNE
Redaktor naczelny prof. dr Leon ŁUKASZEWICZ

Doc. dr hab. inż. Konrad FIAŁKOWSKI (zast. redaktora naczelnego), Władysław KLEPACZ, dr Antoni MAZURKIEWICZ, inż. Dorota PRAWDZIC (zast. redaktora naczelnego), dr inż. Andrzej TARGOWSKI

Sekretarz Redakcji mgr Wanda KAČER **Redaktor techniczny** Bogdan DROZDOWSKI

RADA PROGRAMOWA

Mgr inż. Jan Bursche, mgr inż. Henryk Chyrek, (wiceprzewodniczący) mgr inż. Ryszard Dąbrówka, mgr inż. Bolesław Gliksman, mgr inż. Józef Knysz, prof. dr Leon Łukaszewicz, mgr inż. Jan Matejak, prof. dr Tadeusz Peche (przewodniczący), mgr inż. Jerzy Trybulski (wiceprzewodniczący), dr Tadeusz Walczak, mgr Tadeusz Wasilewski, mgr Waldemar Wiśniewski (sekretarz), mgr Stefan Wojciechowski, dr inż. Henryk Woźniacki, mgr inż. Jan Zdzisław Żydowo

Redakcja: Warszawa, ul. Emilii Plater 20 m. 15, tel. 21-13-91. Zastępca redaktora naczelnego tel. 28-37-29

Zakład Kolportażu WCT NOT, Warszawa, ul. Mazowiecka 12

Zakł. Graf. „Tamka". Z. 2. Zam. 655 Papier powlekany V kl. 80 g. Obj. 3 ark. druk. Nakład 2900. K-59

Cena egzemplarza zł 8.— **INDEKS 36707** **Prenumerata roczna zł 96.—**

maszyny matematyczne

zastosowania w gospodarce, technice i nauce

Nr 11

MIESIĘCZNIK

1 9 7 0

R O K VI

Listopad

Organ Pełnomocnika Rządu do Spraw Elektronicznej Techniki Obliczeniowej i Polskiego Komitetu Automatycznego Przetwarzania Informacji Naczelnej Organizacji Technicznej

POLITECHNIKA ŚLĄSKA BIBLIOTEKA GLIWICE

P. 1877/70

JERZY LESZCZYŃSKI

Centralny Ośrodek Badań i Rozwoju Techniki Kolejnictwa
Warszawa

681.322.06

Dr inż. Jerzy Leszczyński ukończył w roku 1955 studia na Wydziale Komunikacji Politechniki Warszawskiej. Pracuje w dziedzinie ETO od roku 1960 jako kierownik Pracowni, a obecnie Zakładu ETO Centralnego Ośrodka Badań Rozwoju i Techniki Kolejnictwa. Jest autorem szeregu opracowań z zakresu automatyzacji programowania maszyn UMC-1, UMC-10, ODRA-1204; od roku 1968 jest przewodniczącym sekcji ODRA 1204 Klubu Użytkowników Maszyn Cyfrowych ODRA, działającego w ramach PKAPI.

System eksploatacji EMC ODRA 1204/N

Przedstawiono system eksploatacji EMC ODRA 1204, opracowany dla własnych potrzeb w Instytucie Kolejnictwa w Warszawie i używany od 1968 r. System ten zawiera dwa zasadnicze elementy: system operacyjny i translator języka MIN-ALGOL. W artykule opisano: system operacyjny realizujący współpracę pomiędzy sterowaniem, kanałem operacyjnym i urządzeniami zewnętrznymi EMC ODRA 1204 oraz język zewnętrzny stanowiący podzbiór języka ALGOL 60, zwany MIN-ALGOL (w skrócie MINAL).

1. System eksploatacji maszyny cyfrowej ODRA 1204, będący przedmiotem niniejszego artykułu, powstał w Instytucie Kolejnictwa (COBiRTK) na przełomie lat 1967—1968. Zasadniczym elementem uzasadniającym podjęcie prac nad własnym systemem była konieczność wykonania zadań przewidzianych w latach 1968—1969. Zadania te znacznie przekraczały możliwości kupionego zestawu ODRA 1204 w przypadku wykorzystania języków ODRA-ALGOL oraz MOST 2. Wykorzystanie języka adresów symbolicznych (JAS) nie mogło być brane pod uwagę z powodu braku ludzi i czasu. Łączenie języka JAS z językami MOST 2 i ODRA-ALGOL nie zostało przewidziane w systemie eksploatacji maszyny, dostarczanym przez producenta.

Własny system eksploatacji uwzględnia w znacznym stopniu specyfikę zadań, organizację, wyposażenie, obsadę kadrową, a nawet tradycję konkretnego ośrodka obliczeniowego oraz środowiska z nim współpracującego. Jednocześnie system jest na tyle ogólny i elastyczny, że może być interesujący dla szerszego grona użytkowników maszyny cyfrowej ODRA 1204.

System eksploatacji zawiera dwa zasadnicze elementy:

- System operacyjny, który jest właściwie samodzielnym systemem „bazowym" do budowy i uruchamiania elementów oprogramowania podstawowego (translatory, systemy operacyjne itp.). Cechą charakterystyczną jest zachowanie pełnej swobody w operowaniu językiem symbolicznym programowania (JP) w trybie „konwersacji" operatora z maszyną.

- Translator języka MIN-ALGOL, który pozwala na generowanie bardzo zwartych i efektywnych programów wynikowych przy zachowaniu podstawowych dla ALGOLu ułatwień w formułowaniu programu źródłowego.

Zachowano reprezentację wartości całkowitych w postaci liczb krótkich (24 bity), co dla szeregu specyficznych zagadnień znacznie rozszerza możliwości wykorzystania pamięci maszyny. Ponadto umożliwiono formułowanie fragmentów programu źródłowego w języku symbolicznym (JP), co również zwiększa możliwość wykorzystania szybkości i pojemności maszyny.

Uzyskane rezultaty potwierdziły oczekiwania. Uruchomiony w połowie roku 1968 system zdał egzamin w ciągu dwuletniej eksploatacji.

Dalej zostanie opisany dokładniej system operacyjny oraz język MIN-ALGOL. Należy podkreślić, że jakakolwiek praca w omawianym systemie wymaga wpro-

wadzenia systemu operacyjnego do pamięci maszyny. Translator MIN-ALGOL jest wprowadzany tylko na czas translacji programu źródłowego, a potem może być zniszczony. Dotyczy to również innych translatorów dla języków specjalizowanych wykorzystywanych w Ośrodku.

2. System operacyjny (zwany dalej w skrócie SO) realizuje współpracę pomiędzy sterowaniem, kanałem operacyjnym oraz urządzeniami zewnętrznymi maszyny ODRA 1204.

Współpraca jest organizowana przez operatora lub program przy zachowaniu nadrzędnej roli operatora.

SO jest przeznaczony dla zestawu maszyny ODRA 1204 zawierającego:

- jednostkę centralną z pamięcią 16 k,
- jeden kanał współpracy z urządzeniami zewnętrznymi, do którego przyłączone są wymienione niżej urządzenia,
- jeden czytnik taśmy papierowej,
- jeden perforator taśmy papierowej,
- jeden monitor w postaci elektrycznej maszyny do pisania OPTIMA 527.

SO umożliwia jednoczesną pracę z jednym programem ulokowanym w pamięci maszyny. Językiem służącym do opisu programów użytkowych na poziomie SO oraz do współpracy operatora z maszyną służy JĘZYK PODSTAWOWY (zwany dalej w skrócie JP).

SO jest zrealizowany w postaci zwartego bloku programów, zawierającego:

- pakiet programów reakcji na przerwania,
- translator języka JP,
- pakiet programów egzekucji zleceń,
- pakiet podprogramów standardowych.

Blok programów SO zajmuje 2048 miejsc w pamięci maszyny. Miejsca te są niedostępne dla użytkownika maszyny, wykorzystującego system SO. Obszar pamięci dostępny dla użytkownika adresowany jest od 0 do 14 000.

Pakiet podprogramów standardowych zawiera funkcje standardowe języka ALGOL 60 oraz zbiór funkcji specjalnych współpracujących z programem wynikowym, generowanym przez translator MIN-ALGOL (sięganie do tablic, wołanie procedur itp.).

Język JP zawiera rozkazy, dyrektywy, zlecenia, etykiety i komentarze.

Rozkazy opisują operacje wykonywane przez maszynę. Zbiór rozkazów zawiera w sobie listę instrukcji maszyny ODRA 1204 oraz aparat adresowania symbolicznego.

Dyrektywy wiążą symbole adresowania symbolicznego z odpowiednimi miejscami pamięci maszyny.

Zlecenia są narzędziami współpracy operatora z maszyną. Zlecenia są instrukcjami dla SO i umożliwiają wykonywanie czynności związanych z wprowadzeniem, kontrolą, uruchamianiem i wykonywaniem programów.

Etykiety wiążą symbole adresowania symbolicznego z określonymi miejscami programu.

Komentarze stanowią opisy sekwencji programu pomijane w czasie translacji i wykonania programu.

Rozkaz

⟨rozkaz⟩ ::= ⟨operator⟩ (⟨operand⟩)
⟨operand⟩ ::= ⟨adres⟩ ⟨modyfikator⟩
⟨modyfikator⟩ ::= ’|”|”’|.|.’|.”|.”’| PUSTE
W modyfikacji znak ”’” oznacza B-modyfikację, a ilość znaków ”’” określa numer (adres) B-rejestru. Znak ”.” oznacza P-modyfikację.

Przy jednoczesności występowania B- i P-modyfikacji najpierw wykonywana jest B-modyfikacja, a następnie P-modyfikacja.

⟨adres⟩ ::= ⟨adres bzwgl⟩|⟨adres wzgl⟩|
⟨adres symb⟩|⟨PUSTE⟩
⟨adres bzwgl⟩ ::= ⟨adres oktalny⟩|⟨adres dziesiętny⟩
⟨adres wzgl⟩ ::=± ⟨liczba⟩
⟨adres symb⟩ ::= ⟨adres literowy⟩|⟨adres etykiety⟩
⟨adres oktalny⟩ :: = 0 ⟨liczba oktalna⟩
⟨adres dziesiętny⟩ ::= ⟨liczba⟩
⟨adres literowy⟩ ::= ⟨litera⟩ ⟨liczba⟩
⟨adres etykiety⟩ ::=* ⟨liczba⟩ ± ⟨liczba⟩

W powyższych definicjach ”liczba” jest konwencjonalną liczbą dziesiętną bez znaku, a ”liczba oktalna” jest konwencjonalną liczbą oktalną bez znaku.

Liczby o wartości zero mogą być pomijane. Jeżeli przed liczbą o wartości zero umieszczony jest znak + lub — może on być pominięty również.

Adres oktalny określa miejsce w pamięci maszyny o numerze równym wartości liczby oktalnej.

Adres dziesiętny określa miejsce w pamięci maszyny o numerze równym wartości liczby dziesiętnej.

Adres względny określa miejsce w pamięci maszyny odległe od pozycji rozkazu o ilość miejsc równą wartości liczby dizesiętnej.

Miejsca liczone są w przód dla znaku + lub w tył dla znaku —.

Adres literowy określa miejsce w pamięci odległe od miejsca, które przyporządkowano danej literze, o ilość miejsc równą wartości liczby dziesiętnej.

Miejsca liczone są w przód. Litera w adresie literowym może wystąpić tylko po dyrektywie przyporządkowującej literze miejsce w pamięci maszyny.

Adres etykiety określa miejsce w pamięci odległe od miejsca przyporządkowanego etykiecie o ilość miejsc równą wartości liczby dziesiętnej.

Miejsca liczone są w przód dla znaku + lub w tył dla znaku —. Wartość liczby różna od zera może występować tylko w odniesieniu do etykiet położonych wstecz względem pozycji rozpatrywanego rozkazu (tzn. dla etykiet już zdefiniowanych przez pojawienie się w programie).

Przykłady:

11 ()	23(*44)	(077440077)
20 (0277)	20(199’)	(+100)
30 (/199)	21(A.)	(—999)
21 (+5)	17(*66.)	
22 (A)		

Etykieta jest wyrażeniem o postaci:
etykieta ::= *⟨liczba⟩)
Liczba określa symbol etykiety.

Komentarz jest dowolnym ciągiem liter i cyfr, zakończonym powrotem karetki (C R).

Dyrektywa

⟨dyrektywa⟩ ::= = ⟨litera⟩ (⟨operand⟩) | = ⟨litera⟩)

Litera jest symbolem definiowanym, któremu przyporządkowany jest adres pamięci określony operandem.

Operand ma budowę jak w rozkazie.

W przypadku braku znaku „(” (i operanda PUSTE w dyrektywie, symbolowi definiowanemu przyporządkowywany jest adres odpowiadający pozycji programu, w której pojawiła się dyrektywa.

Przykłady:

= A (100)
= A (K 300)
= A (* 144)
= A)

Zlecenie jest wyrażeniem o postaci:
<zlecenie> : : = : <litera> (<operand>)
Litera jest symbolem operacji, którą należy wykonać korzystając z adresu określonego operandem. Operand ma budowę jak wyżej.

Dla zleceń nie posługujących się adresem operand jest obojętny, a może być PUSTE.

Przykłady:

: S (8000)
: K (A 13)
: F (

Zlecenia mogą być wprowadzane z klawiatury monitora lub z taśmy programu. W każdym przypadku następuje wykonanie zlecenia natychmiast po odczytaniu znaku) .

Poniżej podano treść niektórych zleceń.

Symbol zlecenia	Treść zlecenia
G	przejdź do wykonania programu rozpoczynając od wskazanego adresu,
P	powróć do programu,
R	powróć do programu ze zmianą przyłączenia monitor-czytnik perforator,
S	ustaw licznik programu czytającego translatora języka JP na wskazany adres,
T	przygotuj translację,
D	przejdź do dalszego ciągu translacji,
F	wykonaj konsolidację przetranslowanego programu,
K	wydrukuj kontrolnie zawartość komórki pamięci o wskazanym adresie,
Z	wyprowadź kontrolnie liczbę zmiennoprzecinkową z komórki o wskazanym adresie,
X	wczytaj liczbę zmiennoprzecinkową do komórki o wskazanym adresie,
A	rozpocznij translację MIN-ALGOL,
B	rozpocznij wykonywanie programu wynikowego MIN-ALGOL.

Specjalnymi rozkazami są zgłoszenia programu użytkowego skierowane do SO, które powodują przeniesienie sterowania z programu do wnętrza SO. Mają one postać:

6 (N)

gdzie N jest symbolem zgłoszenia.

Poniżej podano przykłady treści zgłoszeń.

Symbol zgłoszenia	Treść zgłoszenia
1	STOP Drukowanie na monitorze tekstu STOP oraz zawartości akumulatora (a), zapamiętanie śladu, włączenie programu czytającego translatora JP.
3	WŁĄCZ MONITOR Włączenie monitora, powrót do programu.
4	WŁĄCZ CZYTNIK — PERFORATOR Włączenie czytnika — perforatora, powrót do programu.
14	CZYTAJ LICZBĘ Z-PRZEC. Czytanie liczby zmiennoprzecinkowej zakończonej "," do a w.
15	CZYTAJ ZNAK Czytanie do a znaku w kodzie BCD, powrót do programu.
21	WYPROWADŹ LICZBĘ KRÓTKĄ Wyprowadzenie zawartości a jako liczby całkowitej 7-pozycyjnej, powrót do programu.
22	WYPROWADŹ LICZBĘ Z-PRZEC. Wyprowadzenie liczby zmiennoprzecinkowej spod adresu wg a oraz wzorca z rejestru w, powrót do programu.
25	WYPROWADŹ ZNAK Wyprowadzenie znaku z a, w kodzie BCD, powrót do programu.

3. Językiem zewnętrznym systemu jest podzbiór języka ALGOL 60, zwany dalej MIN-ALGOL albo w skrócie MINAL.

Język MINAL operuje alfabetem dostosowanym do postaci urządzeń przygotowania taśmy maszyn ODRA 1204, którymi są elektryczne maszyny piszące OPTIMA 527.

Alfabet składa się z liter, cyfr, znaków i symboli słownych:

- litery bez zmian w stosunku do języka wzorcowego ALGOL 60
- cyfry jak wyżej
- znaki działań arytmetycznych + — * /
- znaki inne . : = ; , : ( ) [ ]
- symbole słowne relacji **less** | **lequal** | **equal** | **nequal** | **greater** | **gequal**
- symbole słowne działań logicznych **not** | **or** | **and**
- symbole słowne wartości logicznych **true** | **false**
- symbole słowne operatorów następstwa **goto** | **if** | **then** | **else** | **for** | **do**
- symbole słowne przerywników **step** | **until** | **while** | **comment**
- symbole słowne nawiasów **begin** | **end**
- symbole słowne mian i specyfikacji **boolean** | **integer** | **real** | **array** | **procedure** | **value**

Język MINAL ma ponadto ustalone (nie wymagające deklarowania) nazwy EXP ABS SQRT SIN COS ARCTAN LN SIGN ENTIER read print write code line stop.

W celu uwypuklenia cech składni języka MINAL przedstawiono poniżej ograniczenia w oparciu o opis języka wzorcowego ALGOL 60 [1] nawiązując do odpowiednich paragrafów tego opisu.

2.1. Zbiór wymienionych w alfabecie MINAL liter w stosunku do języka wzorcowego ALGOL 60 bez zmian.

2.2. Zbiór cyfr alfabetu MINAL w stosunku do języka wzorcowego ALGOL 60 bez zmian.

2.3. Zbiór operatorów powstaje ze zbioru operatorów języka wzorcowego przez:

- skreślenie operatorów arytmetycznych ÷ ↑
- skreślenie operatorów logicznych ≡ ⊂
- skreślenie nawiasów)(
- skreślenie mian **own switch**
- skreślenia specyfikatorów **string label**
- skreślenie przerywników 10 ␣
- zastąpienie operatorów relacji i pozostałych operatorów logicznych symbolami słownymi wymienionymi w alfabecie MINAL

2.4. Nazwy w języku MINAL tworzone tylko z DUŻYCH LITER i ewentualnie cyfr. Długość nazw dowolna. Identyfikacja poprzez CZTERY początkowe pozycje nazwy z POMINIĘCIEM SPACJI.

2.5. W języku MINAL nie dopuszcza się zapisu liczb w postaci wykładniczej.

2.6. W języku MINAL nie istnieją łańcuchy w ogólnej postaci.

3.1. Bez zmian.

3.2. W języku MINAL nakłada się ograniczenia na postać parametrów aktualnych procedury, a mianowicie: łańcuch, nazwa przełącznika, nazwa procedury i etykieta nie mogą być parametrem aktualnym procedury.

---

[1] Tekst polski: Stefan Paszkowski — JĘZYK ALGOL 60, PWN, Warszawa 1965, Dodatek A — Opis języka wzorcowego.

3.3. W języku MINAL nie istnieją warunkowe wyrażenia arytmetyczne.

3.4. Zgodnie z 2.3. symbole operatorów relacji i operatorów logicznych zastępuje się symbolami słownymi. Podobnie jak w 3.3. nie istnieją warunkowe wyrażenia boolowskie.

3.5. W języku MINAL jedynym wyrażeniem mianującym jest etykieta. Etykieta musi być nazwą. Nie istnieją przełączniki, przełączenia, warunkowe wyrażenia mianujące.

4.1. Składnia języka MINAL w zakresie instrukcji złożonych i bloków bez zmian w stosunku do składni języka wzorcowego ALGOL 60.

4.2. Składnia instrukcji podstawienia języka MINAL jak wyżej.

4.3. W języku MINAL istnieje tylko instrukcja skoku wg etykiety.

4.4. Składnia języka MINAL w zakresie instrukcji pustych bez zmian w stosunku do składni języka wzorcowego ALGOL 60.

4.5. Składnia instrukcji warunkowych języka MINAL jak wyżej.

4.6. Instrukcja **for** w języku MINAL może posiadać tylko jeden element wykazu **for**. Instrukcja **for ... step** jest określona wyłącznie dla wyrażeń określających podstawienie, krok i granicę typu **integer**.

4.7. Te same ograniczenia postaci instrukcji procedury w zakresie zbioru parametrów aktualnych co w punkcie 3.2. Rodzaj i typ parametru aktualnego musi być zgodny z rodzajem i typem parametru formalnego. Nie dopuszcza się używania procedur rekursywnych. Opis ciała procedury musi być ujęty w nawiasy logiczne **begin ... end;**

5.1. W opisach zmiennych języka MINAL nie występuje typ lokalny własny.

5.2. Opis tablic w języku MINAL podlega ograniczeniom wykluczającym dynamiczne rezerwowanie miejsc na tablice. Każda tablica winna mieć wymieniony wykaz par granicznych; dolną granicą i górną granicą jest liczba dodatnia bez znaku.

5.3. W języku MINAL nie istnieją przełączniki.

5.4. W języku MINAL zbiór specyfikacji procedury nie może zawierać łańcuchów, przełączników, procedur i etykiet. Zbiór specyfikacji winien w pełni odpowiadać zbiorowi parametrów formalnych zarówno pod względem ilości jak i KOLEJNOŚCI parametrów formalnych.

W ciele procedury wolno używać tylko nazw opisanych.

W e j ś c i e jest oznaczone symbolem słownym read występującym w wyrażeniach o postaci:

<nazwa> read ({ * / PUSTE } { m / c / PUSTE });

w którym nazwa określa adres miejsca, na które odbywa się podstawienie wprowadzonej wartości.

<nazwa> : : = <nazwa zmiennej prostej> |
<nazwa zmiennej ze wskaźnikami>

Symbol * oznacza uzależnienie wykonania wprowadzenia wartości od położenia klucza 3 w systemie SO.

Symbol m oznacza wprowadzenie z monitora.

Symbol c oznacza wprowadzenie z czytnika.

Brak symbolu m lub c oznacza wprowadzenie z urządzenia aktualnie przyłączonego, tzn. monitora lub czytnika.

Wartości są zawsze czytane zmiennoprzecinkowo, a następnie przeliczane wg typu nazwy.

Dopuszcza się zapis czytanych wartości w postaci całkowitej i ułamkowej z kropką pozycyjną.

W y j ś c i e jest oznaczone symbolem słownym print występującym w wyrażeniach o postaci:

<wyrażenie ar> print ({ * / PUSTE } { m / p / PUSTE } { — / PUSTE } <wzorzec> ;

w którym wyrażenie ar jest dowolnym wyrażeniem arytmetycznym, którego wartość jest wyprowadzana. Symbol * ma znaczenie identyczne jak w wyrażeniu zawierającym read.

Symbole m i p oznaczają odpowiednio wyprowadzenie wartości na monitor lub perforator lub też w braku tych symboli na urządzenie aktualnie przyłączone.

Symbol „—" oznacza wyprowadzenie z nowej linii. Wartość jest zawsze przeliczana do postaci zmiennoprzecinkowej i wyprowadzana w sposób określony postacią wzorca.

<wzorzec> : : = <człon 1> <człon 2> <człon 3>

Każdy człon jest dwupozycyjną liczbą całkowitą dziesiętną bez znaku:

człon 1 — określa ilość cyfr części całkowitej wyprowadzanej liczby, poprzedzoną spacjami dla pozycji zerowych,

człon 2 — określa ilość cyfr części ułamkowej wyprowadzanej liczby,

człon 3 — określa ilość spacji po wyprowadzonej liczbie.

Dla człon 1 = 0 — liczba jest drukowana w postaci wykładniczej o mantysie 11-pozycyjnej.

Dla człon 1 ≠ 0 oraz człon 2 ≠ 0 — przed częścią ułamkową liczby jest wyprowadzany znak kropka pozycyjna.

Dla człon 1 ≠ 0 oraz człon 2 = 0 — liczba drukowana w postaci całkowitej.

Przy większej ilości cyfr części całkowitej od wartości członu 1 wzorca liczba jest wyprowadzana w postaci wykładniczej.

Przed liczbą wyprowadzany jest znak minus dla wartości ujemnej lub spacja dla wartości dodatniej.

Z m i a n a l i n i i jest oznaczona symbolem line występującym w wyrażeniach o postaci:

line ({ * / PUSTE } { m / p / PUSTE } <liczba>

Symbole * m p mają znaczenie identyczne jak w opisanych wyżej wyrażeniach zawierających read print, a liczba określa ilość linii.

W y p r o w a d z e n i e t e k s t u jest oznaczone symbolem write występującym w wyrażeniach o postaci:

write ({ * / PUSTE } { m / p / PUSTE } ; <tekst> ;

Symbole * m p mają znaczenie jak wyżej.

T e k s t — dowolny ciąg znaków nie zawierający średnika, który zostaje powtórzony przy wykonaniu programu razem z ewentualnie zawartymi w nim znakami spacji i zmian linii.

J ę z y k JP może być używany do opisu ciał procedur przez wykorzystanie symbolu code występującego w przedstawionym niżej kontekście.

<typ> **procedure** <nazwa> (<wykaz par formalnych>)

<wykaz specyfikacji>;

begin code;

tekst programu w języku JP;

end;

Wszystkie jednoliterowe nazwy występujące w wykazie parametrów formalnych procedury mogą być używane w tekście programu opisującego w języku JP ciało procedury jako adresy symboliczne parametrów formalnych.

Wartość procedury jest nadawana przez pobranie do akumulatora (lub akumulatora i wydłużenia) odpowiedniej wartości ostatnim rozkazem w ciele procedury ustalającym stan rejestrów warunków.

Spośród wyrażeń języka JP nie dopuszcza się używania do opisu ciał procedur symbolicznego zapisu etykiety w postaci *<liczba>). Wszystkie przejścia w opisie ciała procedury muszą być określone adresami względnymi.

Zatrzymanie wykonania programu wynikowego jest realizowane symbolem stop występującym w wyrażeniu:

stop $\left(\left\{\begin{matrix}*\\ \text{PUSTE}\end{matrix}\right\} <\text{liczba}>\right)$;

Liczba — jest numerem zatrzymania podanym w postaci liczby całkowitej bez znaku. Wykonanie instrukcji stop powoduje przerwanie wykonywania programu, wydrukowanie na monitorze hasła STOP, a następnie numeru zatrzymania oraz przejścia do programu czytającego języka JP w systemie SO.

Wznowienie wykonywania programu od miejsca przerwania następuje po wykonaniu określonych czynności w systemie SO.

BOLESŁAW GLIKSMAN
ZETO Katowice

681.322.06

Struktura, działanie, oprogramowanie elektronicznej maszyny cyfrowej MIŃSK-32

Część II

SOFTWARE

W drugiej części artykułu przedstawiono organizację wykonywania programów za pomocą systemu DYRYGENT i omówiono bibliotekę programów, system symbolicznego kodowania, strukturę programu symbolicznego oraz translator kodowania operacji polskimi adresami symbolicznymi KOMPAS, opracowany w ZETO-Katowice.

DYRYGENT

Zadania

Organizacja wykonywania programów jest bardzo złożona. Przydział pól pamięci dla poszczególnych programów, jak również sterowanie przebiegiem programów dokonywane są za pomocą systemu DYRYGENT.

W pamięci operacyjnej maszyny znajduje się w czasie jej pracy odpowiednia część DYRYGENTA. Niektóre funkcje programu wypełnia DYRYGENT, a inne bezpośrednio maszyna. Z tego powodu DYRYGENT może być uważany jako programowe rozwinięcie struktury maszyny.
W skład systemu DYRYGENT wchodzi kilka programów organizacyjnych:

Program Wywołanie Początkowe przeznaczony jest do przeniesienia z taśmy magnetycznej do pamięci operacyjnej programów rezydent i koordynator oraz tabeli urządzeń zewnętrznych. Program rezerwuje komórki pamięci dla licznika czasu wykonywania programów.

Program Rezydent przeznaczony jest dla kontroli przechowywania programów kompatibilator i koordynator w pamięci operacyjnej i dla wywołania tych programów w miarę potrzeby. Program rezydent znajduje się stale w pamięci obok jednego z wymienionych programów.

Program Koordynator przeznaczony jest do sterowania równoczesną pracą programów użytkowych. Wywoływany jest do pamięci programowo lub przez operatora maszyny. Osobną częścią koordynatora jest specjalny podprogram nazwany **załadowywacz**, który rozlokowuje program użytkowy w pamięci.

Program Kompatibilator przeznaczony jest do sterowania pracą programów napisanych w języku maszyny MIŃSK-22.

Podobnie jak koordynator może być wywołany programowo lub przez operatora maszyny.

Program Rozładowanie Taśmy Wyjścia przeznaczony jest do wyprowadzania informacji na drukarkę lub perforator kart w przypadku, gdy wobec niesprawności tych urządzeń w czasie wykonywania programu użytkowego, wyniki zapisywane były automatycznie na taśmę. Program ten wykonywany jest na żądanie operatora.

Program Zapis Pakietu na Taśmę Magnetyczną przeznaczony jest dla zapisu pakietu programów użytkowych, wyperforowanych na taśmie lub na kartach, na taśmę magnetyczną. Realizacją programu zapisu steruje program koordynator.

BIBLIOTEKA PROGRAMÓW

Przeznaczenie

Programy użytkowe mogą być pisane, jak dotąd tylko w języku symbolicznego kodowania, którego translator został opracowany przez Ośrodek Naukowo-Badawczy Elektronicznej Techniki Obliczeniowej w Mińsku. System programowania pozwala również na tworzenie innych translatorów. Rezultatem pracy translatora jest program przetłumaczony na język załadowczy. Program w języku załadowczym włączony jest do Biblioteki Programów.

Organizacja Biblioteki Programów realizowana jest za pomocą systemu programów, w skład którego wchodzą:

Komplet Programów Bibliotekarz przeznaczony do wykonywania funkcji operatora taśmy magnetycznej systemu programowania oraz taśm magnetycznych systemów użytkowych.

Program Kompilator przeznaczony do scalania osobno tłumaczonych programów systemu.

Komplet Programów Organizacja Wymiany Informacji z Urządzeniami Podstawowego Zestawu, przeznaczony do wyszukiwania masywów danych, wprowadzania i wyprowadzania danych na kartach i taśmach perforowanych i na taśmie magnetycznej oraz dla kontroli informacji i sygnalizowania błędów.

Program Korektor pozwalający na obróbkę tekstów zapisanych na taśmie magnetycznej.

Program Wprowadzania Programów w języku załadowczym na karty, taśmę perforowaną i drukarkę.

Program Kopiowania Informacji pozwalający na przeniesienie danych z jednego nośnika informacji na drugi.

Program Sortowania Informacji przeznaczony do porządkowania rekordów o stałej długości.
Pakiet Programów Wewnętrznej Obróbki Informacji pozwalający na zmianę kodu, zmianę systemu liczbowego, zmianę kompozycji lub zmianę realizacji.

Pakiet Programów Funkcji Elementarnych.

Język załadowczy jest to język wewnętrzny systemu i w zasadzie nie jest przeznaczony dla celów programowania użytkowego. Program w języku załadowczym jest zbliżony do programu w kodzie maszyny z tą różnicą, że jest niezależny od miejsca zlokalizowania w pamięci operacyjnej. Program w tym języku zawiera obok kodów rozkazów i stałych dodatkowe informacje o rozmieszczaniu jego elementów w pamięci operacyjnej i budowie adresów oraz powiązaniach z programami współpracującymi.

SYSTEM SYMBOLICZNEGO KODOWANIA (SSK)

W skład systemu wchodzi:

Język Symbolicznego Kodowania (JaSK)

Translator Symbolicznego Kodowania (TSK)

Język symbolicznego kodowania stanowi zorientowany maszynowo język dla zapisu programów. Programy w tym języku nazywają się Symbolicznymi Programami (SYMP).

Translator Symbolicznego Kodowania tłumaczy symboliczny program na program w języku załadowczym. Tak więc język symbolicznego kodowania jest odniesiony do języka maszyny w stosunku „jeden do jednego". Poza tym JaSK zawiera możliwości sterowania przebiegiem translacji i realizacją programu jak również zapisu danych i stałych liczbowych lub alfanumerycznych.

Elementy języka

1. Znaki zawarte w normie GOST 10859-64 od 1 do 78. Są to:

a) duże litery alfabetu rosyjskiego

b) pozostałe duże litery alfabetu łacińskiego różniące się od rosyjskiego

c) cyfry

d) znaki + — : , / 10 ↑ = ( ) ; [ ] . ‘ ’ ≠ < > * ␣

2. Etykiety są to symboliczne nazwy nadawane stałym, zmiennym masywom danych i operatorom. Operatory są to rozkazy odpowiadające kodowi maszyny oraz rozkazy podziału pamięci i operacyjne. Etykiety mogą składać się z symboli od 1 do 5, przy czym pierwszym symbolem musi być zawsze litera. Etykiety dzielą się na względne, wartość których zmienia się przy przemieszczeniu programów w pamięci i bezwzględne, wartość których jest stała jak na przykład etykieta numeru symbolu lub liczby elementów masywu danych. Podczas translacji każdej etykiecie nadaje się wartość liczbową.

3. Adresy. Mogą one występować w postaci:

a) etykiet

b) względnych numerów

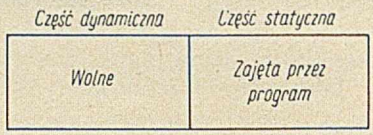
Translator przekształca adres z postaci symbolicznej na bezwzględną w postaci numeru bazy przydzielonego segmentowi programu i adresu odniesionego do tejże bazy.

4. Literały. Występują one jako argumenty operacji zapisane nie w postaci adresów, lecz w postaci liczb lub stałych tekstowych. Literały liczbowe zapisuje się jako liczby stało- lub zmiennoprzecinkowe ze znakiem + lub —. Znak odróżnia literał od adresu. Np.: + 3.47, —$5._{10}$—8, +0.75 Ф, —0.55_{10} —2ФД, +49, +365Д, — 7166B.

Literały tekstowe zapisuje się w postaci informacji alfanumerycznej nie przekraczającej pięciu symboli zamkniętych znakami''. Np.: ‘(TEKST)’,‘(2.500)’,‘(SEK).’

STRUKTURA PROGRAMU SYMBOLICZNEGO

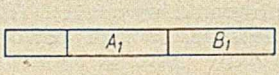
Segment pamięci, do którego wprowadza się programy dzieli się na dwie części



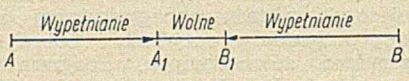
Translator przydziela dla programu część statyczną.

Po przetłumaczeniu otrzymuje się informację o najmniejszej ilości płatów pamięci (po 512 komórek) niezbędnej do realizacji segmentu. Pozostała część pamięci, zarezerwowana do wykonania segmentu jest to część dynamiczna.

Statyczna część każdego segmentu podzielona jest na obszary: podstawowy, roboczy i ogólny. Wymiar każdego obszaru i łączność wzajemną ustala się przy translacji. Obszar podstawowy zawiera rozkazy, stałe i pola i jest ograniczony do 2048 komórek. Obszar ogólny zawiera stałe i pola, w celu wykorzystywania ich w kilku programach, obszar roboczy — tylko pola. Wymiary tych obszarów są ograniczone tylko pojemnością pamięci operacyjnej. Program może zajmować tylko jeden obszar podstawowy lub dowolną ilość różnych obszarów. Segment ładuje się na zasadzie dwustronnego zapełniania pamięci.



Odcinek A_1 B_1 sygnalizowany jest przez wskaźnik zapełnienia pamięci



Przy dynamicznym wykorzystaniu pamięci jeśli na odcinku A_1—B_1 wykorzystuje się N komórek musi być spełniona zależność

$$B_1 - (N+1) > A_1$$

POLSKI TRANSLATOR KOMPAS

Operatory SSK złożone są z liter alfabetu rosyjskiego w układzie stanowiącym skrót nazw w języku rosyjskim.

GRUPA 1 OPERATORY OPISU I ROZDZIAŁU PAMIĘCI

SSK	KOMPAS	ADRESY	NAZWA OPERACJI	SSK	KOMPAS	ADRESY	NAZWA OPERACJI
BAZ	BAZA	N;WYR	BAZA	OPP	OP	VVVVV	OPIS PROGRAMU
NAC	LOK	A1	LOKALIZUJ	OPPS	OPS	VVVVV	OPIS PROGRAMU SPECJALNY
RZB	RZW	N	REZERWUJ	OPPZ	OPZ	VVVVV	OPIS SEGMENTU
RIP	RPX	N	REZERWUJ POLE INDEKSÓW	PUST	LZE	A1;A2	LISTA ZWOLNIONYCH ETYKIET
KW	SL	N	STAŁA LICZBOWA	ZNAC	ZNACZ	N	ZNACZENIE ETYKIETY
KT	ST	VVVVV	STAŁA TEKSTOWA	EKW	EKW	A1	EKWIWALENTNA ETYKIETA
KI	SX	A1;A2	STAŁA INDEKSOWA	PAWP	PAWP	—	ADRES WSKAŹNIKA POMIĘCI
KA	SA	A1;A2	STAŁA ADRESOWA				

GRUPA 2 OPERATORY ROZKAZY
OPERATORY DZIAŁAŃ ARYTMETYCZNYCH I LOGICZNYCH
MODYFIKACJE DZIAŁAŃ

NR MOD	ARGUMENTY DZIAŁAŃ	ZAPIS WYNIKU	KOD MODYFIKACJI SSK	KOD MODYFIKACJI KOMPAS	NR MOD	ARGUMENTY DZIAŁAŃ	ZAPIS WYNIKU	KOD MODYFIKACJI SSK	KOD MODYFIKACJI KOMPAS
0	(A2) * (A1)	A2	3	Z	2	(SM) * (A1)	A2	B	PZ
1	(A2) * (A1)	SM			3	(SM) * (A1)	SM	P	P

SSK	KOMPAS	NAZWA OPERACJI	SKOKI	Z	B/P	SSK	KOMPAS	NAZWA OPERACJI	SKOKI	Z	B/P
		KOD MASZYNY						KOD MASZYNY			
CΦ	DOS	DODAJ STAŁY	+10	+11	+12	ŁC	LDO	LOGICZNIE DODAJ	+24	+25	+26
CΠ	DOZ	DODAJ ZMIENNY	+14	+15	+16	PC	DOP	DODAJ POZYCYJNIE	+04	+05	+06
BΦ	ODS	ODEJMIJ STAŁY	+20	+21	+22	CΦД	DOSD	DODAJ DZIESIĘTNIE			+01
BΠ	ODZ	ODEJMIJ ZMIENNY	+24	+25	+26	BΦД	ODSD	ODEJMIJ DZIESIĘTNIE			+02
УΦ	MNS	MNOŻ STAŁY	+30	+31	+32	УΦД	MNSD	MNOŻ DZIESIĘTNIE			+03
УП	MNZ	MNOŻ ZMIENNY	+34	+35	+36	УЦД	MNCD	MNOŻ CAŁKOWITE DZIESIĘTNIE			+77
ДΦ	DZS	DZIEL STAŁY	+40	+41	+42	УЦ	MNC	MNOŻ CAŁKOWITE			+70
ДП	DZZ	DZIEL ZMIENNY	+44	+45	+46	ДФО	DZR	DZIEL DLA RESZTY			+71
BMΦ	OMS	ODEJMIJ MODUŁY STAŁY	+50	+51	+52	CПОР	DOCH	DODAJ CECHY			+72
BMП	OMZ	ODEJMIJ MODUŁY ZMIENNY	+54	+55	+56	BПОР	ODCH	ODEJMIJ CECHY			+73
ЛСД	LPR	LOGICZNIE PRZESUŃ	+60	+61	+62	СЦКЛ	DOCKL	DODAJ CYKLICZNIE			+74
АСД	APR	ARYTMETYCZNIE PRZESUŃ	+64	+65	+66	НОРМ	NORM	NORMALIZUJ			+75
ЛУ	LMN	LOGICZNIE MNOŻ	+70	+71	+72	СЕ	SUJ	SUMUJ JEDYNKI			+76

OPERATORY PRZESYŁANIA

SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ.	SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ.
П	P	X;A1;A2	PRZEŚLIJ	-10	О	ZAM	X,A1,A2	ZAMIEŃ	-15
ПОЗН	PPZN	X,A1,A2	PRZEŚLIJ Z PRZEC. ZN.	-11	ППОР	PCH	X,A1,A2	PRZEŚLIJ CECHĘ	-16
ПМОД	PMOD	X,A1,A2	PRZEŚLIJ MODUŁ	-12	ГРУП	GRUP	X,A1	GRUPOWY REŻYM	-17
ПЗН	PZN	X,A1,A2	PRZEŚLIJ ZNAK	-14					

OPERATORY SKOKOWE

SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ.	SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ.
ИЗ	SKZ	X;A1;A2	SKOCZ I ZAPISZ	-30	ИНРН	SKNZR	X,A1	SKOCZ PRZY NIERÓWNYM ZERO	-34
ИЧ	SKCZ	X,A1,A2	SKOCZ I CZYTAJ	-36	ИНС	SKNZC	X,A1,A2	SKOCZ PRZY NIEZGODNOŚCI	-37
ИЗН	SKZN	X,A1,A2	SKOCZ WG ZNAKU		ИС	SKL	X,A1,A2	SKOCZ WG LICZNIKA	-22
ИПЛ	SKPL	X,A1	SKOCZ PRZY PLUSIE	-32	ИПЕР	SKN	X,A1,A2	SKOCZ WG WSKAŹN NADMIAR.	-33
ИМН	SKMN	X,A1	SKOCZ PRZY MINUSIE		3	Z	X,A1	ZAPISZ	-30
ИН	SKZR	X,A1,A2	SKOCZ WG ZERA	-34	Ч	CZ	X,A1	CZYTAJ	-36
ИРН	SKRZR	X,A1	SKOCZ PRZY RÓWNYM ZERO						

OPERATORY DZIAŁAŃ NA INDEKSACH

SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD M -20 WYRÓŻNIK	SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD M -20 WYRÓŻNIK
СИ	DOX	X;A1	DODAJ DO INDEKSU	0001	ПАИ	PAX	X;A1	PRZEŚLIJ ADRES POINDEKSU	1001
ВИ	ODX	X;A1	ODEJMIJ OD INDEKSU	0010	ПИА	PXA	X,A1	PRZEŚLIJ INDEKS POADRESU	1010
УИ	MNX	X;A1	MNOŻ INDEKS	0011	ЧИ	CZX	X	CZYTAJ INDEKS	1011
ЛСИ	LDOX	X;A1	LOGICZNIE DODAJ DO IND.	0101	ПА1И1	PA1X1	X;A1	PRZEŚLIJ CZĘŚĆ	1100
ЛУИ	LMNX	X;A1	LOGICZNIE MNOŻ INDEKS	0111	ПА2И2	PA2X2	X;A1	ADRESU($\frac{1}{2}$) DO CZĘŚCI	1101
ПСИ	DOPX	X;A1	DODAJ POZYCYJNIE DOIND.	0110	ПАИ2	PA1X2	X,A1	INDEKSU($\frac{1}{2}$)	1110
ЗИ	ZX	X	ZAPISZ DO INEKSU	1000	ПА2И2	PA2X2	X,A1		1111

OPERATORY DZIAŁAŃ NA SYMBOLACH

SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ	SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ.
ЧС	CZZ	X;N;A1	CZYTAJ SYMBOL	-01	МАС	MAZ	X, A1	MODYFIKUJ ADRES SYMBOLU	-20
ЗС	ZZ	X;N;A1	ZAPISZ SYMBOL	-02					

OPERATORY DZIAŁAŃ NA KOMÓRKACH POZIOMU

SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD M -17 WYRÓŻNIK	SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD M -17 WYRÓŻNIK
СУ	DOPO	X;N;A1	DODAJ DO KOMÓRKI POZ.	0001	ЗУ	ZPO	X;N	ZAPISZ DO KOMÓRKI POZ.	1000
ВУ	ODPO	X;N;A1	ODEJMIJ OD KOMÓRKI POZ.	0010	ПАУ	PAPO	X;N;A1	PRZEŚLIJ ADRES DO K.P.	1001
ЛСУ	LDOPO	X;N;A1	LOGICZNIE DODAJ DO KOM.P.	0101	ПУА	PPOA	X;N;A1	PRZEŚLIJ K.P. DOADRESU	1010
ЛУУ	LMNPO	X;N;A1	LOGICZNIE ODEJMIJ OD K.P.	0111	ЧУ	CZPO	X;N	CZYTAJ KOMÓRKĘ POZIOMU	1011
ПСУ	DOPPO	X;N;A1	DODAJ POZYCYJNIE DO K.P.	0110					

OPERATORY ZE SPECJALNYM BAZOWANIEM

SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ	SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ.
УСП	CZSB	X;A1,A2	CZYTAJ ZE SPEC. BAZOW.	-05	ИП	PODPR	X;A1,A2	SKOCZ DO PODPROGRAMU	-31
ЗСП	ZSB	X;A1,A2	ZAPISZ ZE SPEC. BAZOW.	-06	ВЫХ	WROC	X;A1,A2	WRÓĆ Z PODPROGRAMU	-21
ВКОМ	WR	X;A1;A2	WYKONAJ ROZKAZ	-25	ФА	FA	X;A1;A2	FORMUJ ADRESY	-01
ИЗБ	SKZB	X;A1,A2	SKOCZ I ZAMIEŃ BAZĘ	-26					

OPERATORY POMOCNICZE

SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ.	SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ.
УИ$^0_1$	UD$^0_1$	X,IND	USTAW INDIKATORY	-04	SNOP	NNOP	—	NIE NORMALIZUJ WYNIKÓW	-04
POKP	OKR	—	ZAOKRĄGLAJ	-04	NOP	NOP	X;A1	NIE MA OPERACJI	+00
NOKP	NOKR	—	NIEZAOKRĄGLAJ	-04	PNOR	NOR	—	NORMALIZUJ WYNIKI	-04

OPERATORY PROGRAMÓW OBSŁUGUJĄCYCH

SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ	SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ
РАЗД	ROZDZ	VVVVV	ROZDZIAŁ	—	АСЗЩ	LDOOC	X,A1	LOGICZNIE DODAJ DO K.OCHR.	-17
ИУ$^0_1$	SKPR$^0_1$	X,U,A2	SKOCZ WG WSKAŹ. PROCESORA	-33	ПОЗЩ	DOPOC	X,A1	DODAJ POZYCYJNIE DO K.OCHR.	-17
ИУО$^0_1$	SKUIV$^0_1$	X,U,A2	SKOCZ WG WSKAŹ. U IV	-23	ЛУЗЩ	LMNOC	X,A1	LOGICZNIE MNÓŻ KOM.OCHR.	-17
ЗДВ$^0_1$	ZE$^0_1$	—	WŁĄCZ/WYŁĄCZ ZEGAR	-23	ЗЗЩ	ZOC	—	ZAPISZ DO KOMORKI OCHR.	-17
РРС	SYST	—	ZEZWOL NA PRACĘ W SYST.	-23	ПАЗЩ	PAOC	X,A1	PRZESŁIJ ADRES DO K.OCHR.	-17
ЗРС	NSYST	—	ZAKAZ PRACY W SYST.	-23	ПЗЩА	POCA	X,A1	PRZESŁIJ K.OCHR.DOADRESU	-17
ИКЛ$^0_1$	SKKL$^0_1$	X,N,A2	SKOCZ PRZY KLUCZU	-35	ЧЗЩ	CZOC	—	CZYTAJ KOMÓRKĘ OCHRONY	-17
ПРЕР	PWN	—	ZEZWOL NA PRZERWANIA	-04	ППЧАТ	PPULT	X,A1	PRZEŚLIJ Z PULPITU	-13
ЗПРЕ	NPWN	—	NIE ZEZWOL NA PRZERW.	-04	ПУ$^0_1$	PL$^0_1$	IND.N	PRZEŁĄCZ POZIOM	-27
СУО	DOSO	X,A1,A2	DODAJ DO KOMÓRKI SOP	-17	ПУ$^3_3$	PL$^0_1$Z	IND.N	PRZEŁĄCZ POZIOM Z ZAP.	-27
ВУО	ODSO	X,A1,A2	ODEJMIJ OD KOMÓRKI SOP	-17	ПУ$^3_3$	PL$^0_1$P	IND.N IND.A1	PRZEŁĄCZ POZIOM Z POBR.	-27
ЛСУО	LDOSO	X,A1,A2	LOGICZNIE DODAJ DO KOM. SOP	-17	ПУ^{3}ВК	PL$^0_1$WR	IND.N	PRZEŁĄCZ POZ I WYK ROZKAZ	-27
ПСУО	DOPSO	X,A1,A2	DODAJ POZYC DO KOM SOP	-17	ВЗ	WEZ	X,P,A2	WPROWADŹ Z ZAPISEM SS	-67
ЛУУО	LMNSO	X,A1,A2	MNÓŻ LOGICZNIE KOM. SOP	-17	ЫЗ	WWZ	X,P,A2	WYPROWADŹ Z ZAPISEM SS	-67
ЗУО	ZSO	X,A2	ZAPISZ DO KOMÓRKI SOP	-17	ПЫЗ	PWYZ	X,PA$_2$	REZERW.WPR. WYPR. Z ZAP. SS	-67
ПАУО	PASO	X,A1,A2	PRZEŚLIJ ADRES DO K. SOP	-17	ВБЗ	WEBZ	X,P	WPROWADŹ BEZ ZAPISU SS	-67
ПУОА	PSOA	X,A1,A2	PRZEŚLIJ K SOP DO ADRESU	-17	ЫБЗ	WYBZ	X,P	WYPROWADŹ BEZ ZAPISU SS	-67
ЧУО	CZSO	X,A1	CZYTAJ Z KOMÓRKI SOP	-17	ПЫБЗ	PWBZ	X,P	REZERW.WPR.WYPR. BEZ ZAP.SS	-67
СЗЩ	DOOC	X,A1	DODAJ DO KOM.OCHRONY	-17	ВЗЩ	ODOC	X,A1	ODEJMIJ OD KOM.OCHRONY	-17

GRUPA 3 OPERATORY WSPÓŁPRACY Z URZĄDZENIAMI ZEWNĘTRZNYMI
OPERATORY POMOCNICZE ORGANIZUJĄCE WSPÓŁPRACĘ

SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ	WYRÓŻNIK	SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ	WYRÓŻNIK
КНВУ	SNUZ	T,K	STAŁA NUMERU UZ	—	—	ЗВП	NWP	X,T,K	NIE WCHODŹ DO PRZER.	-60	010001
КОСЛ 5	SWSL5	N,A1,L	STAŁA WYMIANY DO SŁOWA	—	—	РВП	WP	X,T,K	MOŻNA WCHODZIĆ DO PRZER	-60	010010
КОСЛ 6	SWSL6	N,A1,L		—	—	ОВРМ	OPWT	N	OPIS WEJŚCIA TERMINU	-60	001001
КОСИ	SWZB	N,A1,L	STAŁA WYMIANY DO SYMBOLU BEZ SYMBOLU KOŃCA	—	—	ОВПА1	OPWMN	—	OPIS WEJŚCIA Z MONIT.	-60	010000
КОСИ 5	SWZB5	N,A1,L		—	—	ВЛВЫП	BLWYK	—	BLOKUJ WYKON PROGRAMU	-60	010011
КОСИ 6	SWZB6	N,A1,L		—	—	ВЕЗУС	BEZSS	—	BEZ SŁOWA STERUJĄCEGO	—	——
КОСВ	SWZR	N,A1,L	STAŁA WYMIANY DO SYMBOLU RAZEM Z SYMBOLEM KOŃCA	—	—	ПСК	PSEUD	U,A1	PSEUDO STAŁA	—	——
КОСВ 5	SWZR5	N,A1,L		—	—	ИУВУЗ	UZSK$^0_1$	X,U,A2	SKOCZ WG WSKAŹN. DZ	—	——
КОСВ 6	SWZR6	N,A1,L		—	—	ПНК	SPRS	T	SPRAWDŹ POCZ SZYKU TM	-60	001111
ЗАКР	RUZ	X,T,K	REZERWUJ UZ	-60	000100	ЧУС	CZSS	T	CZYTAJ SŁOWO STERUJĄCE	-60	001100
ОСВ	ZUZ	X,T,K	ZWOLNIJ UZ	-60	000010	ПУС	PSS	T	PRZEŚLIJ SŁOWO STERUJĄCE	-60	010110
ЖО	O	X,T,K	OCZEKUJ KOŃCA PRACY UZ	-60	000001						

OPERATORY WSPÓŁPRACY Z URZĄDZENIAMI ZEWNĘTRZNYMI

KOD SSK	* KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ	KOD SSK	* KOMPAS	ADRESY	NAZWA OPERACJI	KOD MASZ.
В	WE	X,P,T,K	WPROWADŹ	-67	ПР	PB	X,P,T,K	PRZEPUŚĆ BLOK TM	-67
ВЫ	WY	X,P,T,K	WYPROWADŹ	-67	ПН	PT	X,T,K	PODPROWADŹ POCZĄTEK TM	-67
ВР	WB	X,P,T,K	WRÓĆ O BLOK TM	-67	СН	RT	X,T,K	ROZŁĄDUJ TM	-67

REŻYM WSPÓŁPRACY	BEZ OCZEKIWANIA SSK	BEZ OCZEKIWANIA KOMPAS	Z OCZEKIWANIEM SSK	Z OCZEKIWANIEM KOMPAS
BEZ BADANIA WSKAŹNIKÓW	*	*	* ЖС	* O
Z BADANIEM WSKAŹNIKÓW PO ZAKOŃCZENIU I SKOK	—	—	* ИУ	* SK
Z BADANIEM WSKAŹNIKÓW PRZED ROZPOCZ. I ZAPIS	УЗ *	WZ *	УЗ * ЖС	WZ * O
Z BADANIEM WSKAŹNIKÓW PO ZAKOŃCZENIU I ZAPIS	* УЗ	* WZ	* УЗ ЖС	* WZO
Z BADANIEM WSKAŹN. PRZED ROZP. PO ZAKONCZ. I ZAPIS	* ЗУ	* ZW	* ЗУ ЖС	* ZWO

GRUPA 4 OPERATORY STEROWANIA PRACĄ PROGRAMU

SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD M -60 WYRÓŻNIK	SSK	KOMPAS	ADRESY	NAZWA OPERACJI	KOD M -60 WYRÓŻNIK
ОСТ	STOP	—	STOP	000011	ЖСООС	OOPKS	X,T,K	OCZEKUJ ODPW OPER.SPEC.	010111
ВРМ	CZAS	—	PODAJ CZAS	001110	ЗГР	ZAL	X	ZAŁADUJ SEGMENT	000101
ДАТА	DATA	—	PODAJ DATĘ	001011	ЗСЕГ	ZSEG	X	ZWOLNIJ SEGMENT	001010
ПРМ	DRMN	X,T,K	DRUKUJ NA MONIT.	000111	ПОВТ	POWT	N,A1	POWTÓRZ	001101
ЖООО	OOPR	X,T,K	OCZEKUJ ODPOW. OPER.	001000					

GRUPA 5 OPERATORY STEROWANIA WYDRUKIEM TRANSLACJI

SSK	KOMPAS	ADRESY	NAZWA OPERACJI	SSK	KOMPAS	ADRESY	NAZWA OPERACJI
ЗАГЛ	NAGL	VVVVV	NAGŁÓWEK	ПРОП	PRZEP	N	PRZEPUŚCIĆ
КОММ	KOM	VVVVV	KOMENTARZ				

OZNACZENIA:

X - INDEKS
A1 - ADRES PIERWSZY
A2 - ADRES DRUGI
P - ZNACZNIK OPERACJI
T - TYP URZĄDZENIA ZEWN.
K - NUMER W TYPIE
N - NUMER
IND - INDYKATOR
U - WSKAŹNIK
L - LICZBA KOMÓREK
D - NR SYMBOLU W SŁOWIE
OP - OPIS OBSZARU
V - SYMBOL

Oprogramowanie podstawowe dostarczone przez producenta pozwala na komunikowanie się z maszyną w zdaniach pochodnych języka rosyjskiego wyłącznie w kodach według GOST 10859—64.

Jak wiadomo dostępny w kraju park urządzeń do przygotowania danych oparty jest na alfabecie łacińskim. Duża część personelu, przede wszystkim operatorskiego, nie zna języka rosyjskiego. Wymienione względy skłoniły grupę programistów Zakładu Elektronicznej Techniki Obliczeniowej w Katowicach do opracowania polskiego translatora opartego na systemie symbolicznego kodowania pod nazwą KOMPAS (skrót: Kodowanie Operacji MIŃSKA-32 Polskimi Adresami Symbolicznymi).

KOMPAS charakteryzuje się:

1) znakami alfabetu łacińskiego,

2) polskimi nazwami mnemotechnicznymi operatorów, odpowiadających operatorom rosyjskiego JaSK,

3) dodatkowymi wyżej zorganizowanymi operatorami

4) wprowadzeniem procedur wejścia i wyjścia w różnych kodach taśmy (5 i 8 ścieżek)

5) wprowadzeniem procedur wejścia i wyjścia w różnych kodach kartowych.

Translator KOMPAS będzie współpracować z programami systemu dyrygent i systemu bibliotekarz.

Komunikacja człowieka z maszyną odbywać się będzie za pomocą KOMPAS-u co znacznie ułatwi pracę w polskich Ośrodkach Obliczeniowych.

Operatory języka symbolicznego podzielone są na 5 grup według spełnianych funkcji. W tabelach na str. 7 zestawione są kody, adresy i funkcje wszystkich operatorów JaSK i KOMPAS.

CZESŁAW PACZUŁA
Zakład Badań i Doświadczeń Budownictwa Węglowego
Katowice

Niektóre problemy zastosowania elektronicznej techniki obliczeniowej w budownictwie

Współczesna gospodarka charakteryzuje się wysokim tempem wzrostu produkcji i postępu techniczno-organizacyjnego. Będące do dyspozycji coraz nowocześniejsze środki techniczne produkcji, najnowsze metody wytwarzania, racjonalna organizacja produkcji, ciągły wzrost wydajności pracy, stawiają określone wymagania pod adresem kierownictwa jednostek gospodarczych. Racjonalne zarządzanie i kierowanie współczesnymi przedsiębiorstwami wymaga sprawnego systemu informacji w ramach przedsiębiorstw i ich zjednoczeń. Tylko na podstawie wiarygodnych i szybko dostarczonych aktualnych informacji można podejmować optymalne decyzje. Wymienione okoliczności leżą u podłoża tendencji zmierzających do modernizacji istniejących systemów przetwarzania danych, jako nie odpowiadających potrzebom współczesnej gospodarki. Powoduje to prowadzenie na coraz szerszą skalę prac przygotowawczo-projektowych, poprzedzających zastosowanie elektronicznej techniki obliczeniowej oraz prac wdrożeniowych w tym zakresie.

Problematyka ta uwydatnia się szczególnie w przedsiębiorstwach tych gałęzi gospodarki, w których występuje przepływ dużej masy informacji: w nich też powinno się w pierwszej kolejności modernizować istniejące tradycyjne systemy przetwarzania danych.

O wielkiej wadze i docenianiu tych zagadnień w naszych przedsiębiorstwach budowlano-montażowych i przemysłu materiałów budowlanych, w projektowaniu inwestycji i innych dziedzinach szeroko rozumianego przemysłu budowlanego świadczy zorganizowana w maju br. w Gdyni trzydniowa I Ogólnokrajowa Ekonomiczno-Techniczna Konferencja Naukowa na temat zastosowania elektronicznej techniki obliczeniowej w przemyśle budowlanym.

Organizatorami tej ze wszech miar interesującej konferencji byli:

- Ministerstwo Budownictwa i Przemysłu Materiałów Budowlanych,
- Polskie Towarzystwo Ekonomiczne Oddział Wojewódzki w Gdańsku,
- Centrum Elektronicznej Techniki Obliczeniowej Przemysłu Budowlanego „ETOB",
- Gdańskie Zjednoczenie Budownictwa,
- Zakład Badań i Doświadczeń Gdańskiego Zjednoczenia Budownictwa,

Konferencja odbywała się pod protektoratem Członka Biura Politycznego KC PZPR wicepremiera tow. Stanisława Kociołka oraz ministra Budownictwa i Przemysłu Materiałów Budowlanych tow. Andrzeja Gierszа.

Celem Konferencji było:

- przedstawienie konkretnych wyników dotychczasowych prac w zakresie zastosowań ETO w przemyśle budowlanym,
- szeroka wymiana doświadczeń,
- ustalenie głównych kierunków zastosowań ETO w przemyśle budowlanym w najbliższej przyszłości i w planach perspektywicznych.

Ideą przewodnią Konferencji było z jednej strony przedstawienie dotychczasowego kilkuletniego i dość dużego już dorobku w dziedzinie praktycznego wdrażania ETO w budownictwie i przemyśle materiałów budowlanych, z drugiej zaś wywołanie dyskusji nad sformułowaniem selektywnych kierunków zastosowań ETO w przemyśle budowlanym.

Kierunki te mają stanowić merytoryczną bazę do koncentracji sił i środków pozwalając na szybkie wypracowanie przez nieliczną jeszcze kadrę ETO zunifikowanych programów i systemów o najwyższej

efektywności zastosowań oraz rezygnację z systemów i kierunków o niskim zapotrzebowaniu i słabych walorach eksploatacyjnych.

Konferencja poświęcona była zastosowaniu ETO w przemyśle budowlanym z następującym podziałem tematycznym na 3 grupy problemowe:

I — zastosowanie ETO do bilansowania, optymalizacji planów oraz programowania procesu inwestycyjnego w ramach branży i regionu;

II — Zastosowanie ETO w planowaniu, organizacji i zarządzaniu produkcją i działalnością oraz do ewidencji i rozliczania działalności gospodarczej przedsiębiorstw przemysłu budowlanego;

III — Zastosowanie ETO do obliczeń naukowych, inżynieryjno-technicznych i sterowania procesami produkcji w przemyśle materiałów budowlanych.
Zgodnie z podziałem tematycznym obrady prowadzone były plenarnie oraz w sekcjach.

W każdej grupie problemowej znalazły się szczegółowe referaty, które precyzowały zagadnienie i stanowiły przedmiot dyskusji.

Konferencja potwierdziła szerokie stosowanie elektronicznej techniki obliczeniowej w przemyśle budowlanym na całym świecie. Nasz stosunkowo skromny krajowy park ok. 170 elektronicznych maszyn cyfrowych wykorzystywany jest dla celów problematyki budownictwa w około 50 procentach.

Na rozwój ETO w przyszłej pieciolatce 1971—1975 przemysł budowlany przeznaczył około 1,5 mld zł nakładów, biorąc pod uwagę przede wszystkim efekty gospodarcze, jakie ta nowoczesna technika zapewnia. Kadra fachowców — problemistów, programistów, projektantów systemów, analityków itp. licząca dziś w resorcie około 300 osób, wzrośnie do około 3 000 osób w roku 1975. Dane te świadczą o randze zagadnienia i miejscu, jakie mu wyznaczono w przemyśle budowlanym.

Już obecnie ETO znalazła szerokie zastosowanie w resorcie budownictwa i przemysłu materiałów budowlanych. Świadczą o tym coraz szersze, konkretne zastosowania ETO w wielu jednostkach przemysłu budowlanego. Na specjalne podkreślenie zasługują duże osiągnięcia Gdańskiego Zjednoczenia Budownictwa, Śląskiego Zjednoczenia Budownictwa Mieiskiego w Katowicach (system „NW"), Zjednoczenia Budowlano-Montażowego Przemysłu Węglowego (systemy JSB, ESIT i MID), Poznańskiego Zjednoczenia Budownictwa (system ETOPLAN) i Zjednoczenia Budownictwa „Warszawa" (systemy MID i TRANSPORT).

Wymienione systemy realizowane są aktualnie wyłącznie w ośrodkach obliczeniowych innych resortów, w szczególności sieci ZETO.

W resorcie budownictwa i przemysłu materiałów budowlanych, jednostkę wiodaca w zakresie zastosowań elektronicznej techniki obliczeniowej jest Centrum Elektronicznej Techniki Obliczeniowej Przemysłu Budowlanego „ETOB" w Warszawie.

Centrum „ETOB" posiada obecnie pięć zakładów obliczeniowych zlokalizowanych w Warszawie, Krakowie, Poznaniu, Łodzi i Bydgoszczy. W okresie od roku 1975 przewiduje się wzrost ilości zakładów do 17 oraz powiązanie ich siecią transmisji danych umożliwiających przesyłanie na odległość przetworzonych przez komputery danych. Niezależnie od zakładów organizacyjnie podległych bezpośrednio Centrum ETOB, działalność w zakresie elektronicznego przetwarzania danych prowadzą Zakład Elektronicznej Techniki Obliczeniowej ETOB przy Zjednoczeniu Budownictwa Hutniczego w Katowicach oraz Gdański Ośrodek ETO „GETOB". Ponadto prace związane z przygotowaniem i wdrożeniem ETO prowadzą pracownie i ośrodki przy zjednoczeniach budownictwa i przemysłu materiałów budowlanych jak również w instytutach resortowych i biurach projektów.

Działalność w zakresie ETO w resorcie jest nadzorowana przez Departament Ekonomiki i Finansów Ministerstwa Budownictwa i Przemysłu Materiałów Budowlanych.

Łączna ilość elektronicznych maszyn cyfrowych, które zgodnie z resortowym planem rozwoju ETO oraz krajowym programem rozwoju informatyki zainstalowane zostaną do roku 1975 wyniesie 46 egzemplarzy, w tym 37 maszyn do przetwarzania danych oraz 9 do obliczeń numerycznych.

Możliwości zastosowania nowoczesnej techniki obliczeniowej w budownictwie są bardzo duże dla celów zarządzania, projektowania i realizacji zadań. Kryją się za tym duże możliwości skrócenia cykli budowy, ścisłego powiązania produkcji materiałów budowlanych z wykonawstwem obiektów, znacznego obniżenia kosztów produkcji budowlano-montażowej (8—19%) i optymalizacji produkcji (około 30%). Można się o tym przekonać w niejednym przedsiębiorstwie budowlano-montażowym, które wdrożyło komputery do swojej pracy. Zagadnieniom tym poświęcono wiele miejsca w referatach i ożywionej dyskusji w toku bardzo sprawnie prowadzonych obrad.

W niedalekiej przyszłości, bo jeszcze w bieżącym roku resort budownictwa otrzyma ze Związku Radzieckiego 4 nowoczesne maszyny elektronowe typu „MIŃSK 32", a w ślad za nimi przyjdą wkrótce dalsze.

Na konferencję przygotowano łącznie 29 referatów, w tym 10 na obrady plenarne i 19 na obrady w sekcjach. Obejmowały one szeroki wachlarz zagadnień, począwszy od przedstawienia węzłowych problemów postępu techniczno-organizacyjnego w przemyśle budowlanym w świetle zadań planu pięcioletniego 1971—1975 (referat ministra A. Giersza) po szczegółowe prezentacje konkretnych, różnorodnych doświadczeń praktycznych z zastosowaniem ETO w wykonawstwie i projektowaniu.

Tematyka referatów reprezentowana przez uczestników poszczególnych środowisk była bardzo interesująca. Z tego wiec względu celowe wydaje się przedstawienie chociaż tytułów referatów i ich autorów, pozwoli to najlepiej zobrazować kierunki badań i zastosowań ETO w poszczególnych ośrodkach i dziedzinach przemysłu budowlanego:

POSTEP TECHNICZNO-ORGANIZACYJNY W PRZEMYŚLE BUDOWLANYM W ŚWIETLE ZADAŃ PLANU PIĘCIOLETNIEGO 1971—1975

— Andrzej Giersz — minister Budownictwa i Przemysłu Materiałów Budowlanych

OGÓLNA KONCEPCJA ZASTOSOWAŃ ETO ORAZ KIERUNKI ROZWOJU ETO W PRZEMYŚLE BUDOWLANYM W LATACH 1970—1975

— mgr inż. Andrzej Dąbkowski — Ministerstwo Budownictwa i PMB, (Dep. Ekonomiki i Finansów)

Prezentacja doświadczeń i kierunków zastosowań ETO w Gdańskim Zjednoczeniu Budownictwa

— mgr Bolesław Filar, mgr inż. Jan Kempa, mgr Jan Nowak, mgr Andrzej Pankowski, mgr Bronisław Tyburczy, mgr Stefan Zawadzki — Gdańskie Zjednoczenie Budownictwa

OGÓLNA CHARAKTERYSTYKA ZASTOSOWAŃ ELEKTRONICZNEJ TECHNIKI OBLICZENIOWEJ W ZARZĄDZANIU PRODUKCJĄ BUDOWLANĄ

— mgr Kazimierz Pakulski — Centrum ETOB

NOWE TENDENCJE INTEGRACJI SYSTEMU ZARZADZANIA PRZY ZASTOSOWANIU ELEKTRONICZNEJ TECHNIKI OBLICZENIOWEJ

— doc. dr hab. Janusz Gościński — Centralny Ośrodek Doskonalenia Kadr Kierowniczych

OGÓLNA CHARAKTERYSTYKA ZASTOSOWAŃ ETO DO BILANSOWANIA I OPTYMALIZACJI ORAZ PROGRAMOWANIA PROCESU INWESTYCYJNEGO W RAMACH BRANŻY I REGIONU

— mgr inż. Kazimierz Husarski — Centrum ETOB

DOŚWIADCZENIA ŚLĄSKIEGO ZJEDNOCZENIA BUDOWNICTWA MIEJSKIEGO W KATOWICACH W ZAKRESIE STOSOWANIA ETO W BUDOWNICTWIE MIEJSKIM

— mgr Stanisław Nowak — Śląskie Zjednoczenie Budownictwa Miejskiego w Katowicach

ETO W ZARZĄDZANIU PRODUKCJĄ PRZEDSIĘBIORSTWA BUDOWLANO-MONTAŻOWEGO SYSTEM MID

— dr inż. Jerzy Moliński — Zjednoczenie Budownictwa „Warszawa"

SYSTEM AUTOMATYCZNEGO PRZETWARZANIA INFORMACJI W ZAKŁADACH WYTWÓRCZYCH PREFABRYKATÓW ŻELBETOWYCH (ESPIR-„A")

— mgr inż. Michał Masłowski

SYSTEM ETOPLAN W POZNAŃSKIM ZJEDNOCZENIU BUDOWNICTWA

— mgr inż. Marian Waller — Zakład Badań i Doświadczeń przy Poznańskim Zjednoczeniu Budownictwa

ZINTEGROWANY SYSTEM AUTOMATYZACJI ZARZĄDZANIA PRZEDSIĘBIORSTWEM TRANSPORTOWYM BUDOWNICTWA

PRZYKŁAD KONCEPCJI REALIZOWANEJ W ZAKŁADACH TRANSPORTOWYCH BUDOWNICTWA „WARSZAWA"

— mgr Olgierd Bobrowicz — Zakłady Transportowe Budownictwa „Warszawa"

ZASTOSOWANIE NOWOCZESNYCH METOD OPTYMALIZACJI W PROJEKTOWANIU KONSTRUKCJI INŻYNIERSKICH

— mgr inż. R. Dąbrówka, dr inż. J. Goliński — SOETO

ZASTOSOWANIE ETO W PROJEKTOWANIU KONSTRUKCJI BUDOWLANYCH NA PRZYKŁADZIE KRAKOWSKIEGO BIURA PROJEKTÓW BUDOWNICTWA PRZEMYSŁOWEGO

— mgr inż. Tomasz Bocheński — Krakowskie Biuro Proj. Bud. Przem.

ZASTOSOWANIE ELEKTRONICZNEJ TECHNIKI OBLICZENIOWEJ W KOSZTORYSOWANIU

— inż. arch. Henryk Hajdasz — „Miastoprojekt" Częstochowa

ZASTOSOWANIE ELEKTRONICZNEJ TECHNIKI OBLICZENIOWEJ W PROJEKTOWANIU INSTALACJI SANITARNYCH I ENERGETYCZNYCH

— mgr inż. Hanna Krzyszczuk — Centrum ETOB

ZASTOSOWANIE EMC DO STEROWANIA PROCESAMI TECHNOLOGICZNYMI W PRZEMYŚLE MATERIAŁÓW BUDOWLANYCH

— dr inż. Marek Grochowski — JPSiC Warszawa

ZAGADNIENIE OPTYMALIZACJI PROCESU REGENERACJI CIEPŁA

— mgr inż. Bohdan Kułakowski — Instytut Przemysłu Szkła i Ceramiki

PROBLEMY AUTOMATYZACJI INFORMACJI DOKUMENTACYJNEJ BUDOWNICTWA I PMB

— mgr inż. Zbigniew Staniszewski — Centralny Ośrodek Informacji Budownictwa

PROBLEMY ZASTOSOWANIA INFORMATYKI W PLANOWANIU CENTRALNYM W RESORCIE BUDOWNICTWA I PRZEMYSŁU MATERIAŁÓW BUDOWLANYCH

— dr Józef Oleński — Ministerstwo Budownictwa i PMB (Dep. Ekonomiki Finansów)

WSPÓŁPRACA KRAJÓW CZŁONKOWSKICH RWPG W AUTOMATYZACJI PROJEKTOWANIA BUDOWNICTWA

— mgr inż. Stanisław Rybicki — Centrum Elektronicznej Techniki Obliczeniowej Przemysłu Budowlanego

OPTYMALIZACJA PLANU INWESTYCJI m. KOSZALINA

— mgr inż. Kazimierz Husarski — Centrum ETOB

SYSTEM ROZLICZEŃ I FAKTUROWANIA SPRZĘTU BUDOWLANEGO

— mgr Antoni Kurek — Dział Studiów i Projektów Syst. EPD Centrum „ETOB" ZO — Kraków

SYSTEM EWIDENCJI, ROZLICZEŃ, FAKTURACJI I SPRAWOZDAWCZOŚCI EKSPLOATACYJNEJ W PRZEDSIĘBIORSTWIE TRANSPORTOWYM BUDOWNICTWA

— mgr Janina Skoczek — Centrum Elektronicznej Techniki Obliczeniowej Przemysłu Budowlanego

PROBLEMY FORMOWANIA MODELI OPTYMALNEGO ROZMIESZCZENIA MOCY PRODUKCYJNYCH

— mgr inż. Zdzisław Jasłowski — Instytut Organizacji i Mechanizacji Budownictwa

SYSTEM AUTOMATYCZNEGO PLANOWANIA, LIMITOWANIA I ROZLICZANIA PODSTAWOWYCH ŚRODKÓW PRODUKCJI

— Stefan Gogdaszewski — Centrum ETON Warszawa

SYSTEM EWIDENCJI I ROZLICZEŃ ZATRUDNIENIA I PŁAC W PRZEDSIĘBIORSTWACH BUDOWLANO-MONTAŻOWYCH

— dypl. ek. Izabella Linkowska, dypl. ek. Remigiusz Kwietniewski, Centrum Elektronicznej Techniki Obliczeniowej Przemysłu Budowlanego „ETOB" Warszawa, ul. Bema 65

SYSTEM PLANOWANIA I KONTROLI WYKONANIA PLANU PRODUKCJI ORAZ ZUŻYCIA ŚRODKÓW PRODUKCJI W ZAKŁADACH BUDOWNICTWA PRODUKCJI BETONÓW

— mgr Tadeusz Guzowski, — Zakł. Obliczeniowy ETOB Łódź

W wydanych drukiem, starannie opracowanych materiałach konferencyjnych, oprócz wymienionych problemów przedstawiono także wiele innych spraw dotyczących zastosowań ETO w budownictwie. Nad referatami wywiązała się bardzo żywa i rzeczowa dyskusja, w której udział brali zarówno praktycy jak i teoretycy ETO.

W dyskusji, w której nie brakowało wypowiedzi kontrowersyjnych, a która miała charakter roboczy i szczerej wymiany poglądów, wykazano walory i przydatność praktyczną zaprezentowanych doświadczeń w zakresie zastosowań ETO w przemyśle budowlanym.

Zwrócono również uwagę na pewne niedociągnięcia w pracach organizacyjno-przygotowawczych, systemowych i koordynacyjnych związanych z wdrożeniem ETO w budownictwie.

Uczestnicy konferencji wysunęli wiele ciekawych propozycji zmierzających do udoskonalenia prac w zakresie dalszego rozwoju ETO. Postulowano przede wszystkim skoncentrowanie wysiłku badawczego i prac wdrożeniowych na zagadnieniach:

- określenia właściwych kierunków selektywnego rozwoju ETO w przemyśle budowlanym jako całości,
- zdeterminowania form integracji różnych systemów w poszczególnych ośrodkach i w skali resortu,
- zwrócenia większej uwagi na doskonalenie bazy dokumentacyjnej w budownictwie. Dokumenty stanowią źródło, w oparciu o które sporządzone zostają nośniki informacji, stanowiące dane wyjściowe do maszyn matematycznych. Żadna maszyna nie da poprawnych wyników, jeśli dane wejściowe są błędne.

Dlatego tak decydująca rola przypadnie w nowym systemie dokumentom źródłowym. Dokumenty te muszą być jednolicie i bezbłędnie opracowane, możliwie w sposób zautomatyzowany, przy maksymalnym wykorzystaniu maszyn peryferyjnych,

- przeprowadzenia szerokiego planowego szkolenia kadry specjalistów i kadry kierowniczej przemysłu budowlanego w zakresie metod sieciowych i ETO,
- zwrócenia większej niż dotychczas uwagi na zacieśnienie i rozszerzenie wzajemnej współpracy między ośrodkami badawczymi i wdrażającymi ETO w budownictwie, m. in. uregulowania wymaga sprawa odpłatności wymiany doświadczeń ośrodków różnych resortów,
- dostatecznie wczesnego przygotowania techniczno-organizacyjnego i kadrowego do efektywnego i natychmiastowego wykorzystania przyszłego parku maszyn.

Z doświadczeń niektórych ośrodków obliczeniowych w kraju wynika, że starania o maszynę elektronową trwają nieraz bardzo długo. Mimo to po otrzymaniu maszyny wielomilionowej wartości często ośrodek nie ma przygotowanych programów, kadr itp., a często nawet w pełni wykończonych pomieszczeń.

Uczestnicy postulowali zwoływanie tego typu konferencji corocznie — przy czym celowe wydaje się, aby konferencje przyszłe mniej zajmowały się problemami teoretycznymi (jest coraz więcej publikacji fachowych), a stały się „giełdą ETO" — wymiana konkretnych osiągnięć w zakresie zastosowań ETO w budownictwie.

Ta garść refleksji uczestnika konferencji nie wyczerpuje rzecz jasna problematyki poruszonej zarówno w referatach jak i w dyskusji na konferencji, która zgromadziła ponad 600 fachowców z całego kraju. Już sama liczba uczestników świadczy o ogromnym zainteresowaniu problemami elektronicznej techniki obliczeniowej w naszym kraju.

Można liczyć, że wnioski z konferencji poważnie wzbogacą program wdrażania nowoczesnej techniki do praktyki polskiego budownictwa, które doskonali metody zarządzania, programowania i wykonawstwa. Przy okazji warto wspomnieć, że wśród krajów należących do RWPG i zajmujących się tymi problemami, Polska znajduje się w czołówce. Za kilka miesięcy przewiduje się międzynarodowe seminarium w ramach RWPG, na którym przedstawiciele poszczególnych krajów przedstawią swoje osiągnięcia w tym zakresie oraz podzielą się doświadczeniami.

Dotychczasowe doświadczenia i zamierzenia budownictwa w zakresie stosowania elektronicznej techniki obliczeniowej do celów zarządzania i kierowania produkcją nasuwają jednak szereg wniosków wykraczających poza tę branżę.

Dotychczas w budownictwie — jak już wyżej wspomniano — nie pracuje ani jedna maszyna cyfrowa, choć potrzeby są ogromne. W najbardziej uprzemysłowionym w kraju województwie katowickim pracuje w przemyśle hutniczym zaledwie jedna maszyna dużej mocy, podobna jest też sytuacja w przemyśle węglowym. Do końca przyszłego roku wzbogacimy się prawdopodobnie o kilkanaście maszyn średniej i dużej mocy. Będzie to jednak przysłowiowa kropla w morzu gospodarczych potrzeb. Stąd też nie po raz pierwszy nasuwa się pod adresem czynników planistycznych postulat przyspieszenia rozwoju naszego przemysłu elektronicznego, aby był on zdolny do zaopatrzenia naszej gospodarki i nauki w bardziej sprawne, nowoczesne komputery i odpowiedni dla nich sprzęt peryferyjny. Import gotowych kompleksów urządzeń obliczeniowych nawet o średniej tylko mocy — ze względu na ich wysoki koszt i masowość zapotrzebowania — nie rozwiąże problemu. Pozostaje więc jedyna najbardziej racjonalna droga zakupu odpowiednich licencji i należyte techniczno-organizacyjne i kadrowe przygotowanie się do ich wykorzystania w praktyce przemysłowej. Dyskusje na ten temat toczą się także między państwami — członkami RWPG, lecz na razie bez widocznych rezultatów. A jednak c z a s, ten niedoceniany u nas czynnik ekonomiczny — zmusza do przyspieszenia konkretnych starań o szybsze nasycenie naszej gospodarki w to coraz bardziej niezbędne narzędzie nowoczesnego kierowania i zarządzania.

Szczególnie dotkliwie odczuwają braki w dziedzinie elektronicznej techniki obliczeniowej województwa katowickie i warszawskie. Z kolei właśnie one posiadają znaczne możliwości kadrowe i wytwórcze, aby pokusić się o podjęcie tego trudnego, lecz realnego zadania — stworzenia drugiej po Wrocławiu krajowej bazy wytwórczej urządzeń techniki obliczeniowej średniej mocy.

Powodzenie tego zamierzenia w jednym z tych województw zapewnia bogata sieć wyższych uczelni, placówek naukowo-badawczych i projektowo-konstrukcyjnych, a także stosunkowo liczna już specjalistyczna kadra elektroników i automatyków. Wzbogacenie profilu wytwórczego tych województw o tego rodzaju produkcję byłoby przedsięwzięciem nad wyraz opłacalnym nie tylko dla danego regionu.

UWAGA, PRENUMERATORZY CZASOPISM WCT NOT

Zakład Kolportażu Wydawnictw Czasopism Technicznych NOT począwszy od 1971 roku wprowadza do sposobów prenumerowania czasopism technicznych poważne udogodnienie, które:

— odciąży prenumeratorów,
— usprawni pracę kolportażu,
— spowoduje oszczędności finansowe.

Będzie to tzw. prenumerata ciągła, obowiązująca zakłady pracy, biblioteki, organizacje itp.

Instytucja, która zamówi czasopisma techniczne WCT NOT na 1971 r. i wpłaci należność za ten okres, nie jest obowiązana w latach następnych (1972, 1973, 1974 itd.) nadsyłać co roku nowych zamówień, ponieważ prenumerata ciągła ważna jest na czas nieograniczony. Dla utrzymania abonamentu wystarczy w latach następnych wpłacić w przewidzianym terminie od 1 lipca do 20 listopada należność za prenumeratę na rok następny.

Zamówienia na prenumeratę ciągłą na rok 1971 prosimy nadsyłać w okresie od 1.VII. do 20.XI. br. do Zakładu Kolportażu WCT NOT, Warszawa, ul. Mazowiecka 12, nr konta 1-9-121697 wnosząc jednocześnie należność za jeden rok.

W przypadku jakichkolwiek zmian (tytułów, rezygnacji z prenumeraty itp.) prosimy o natychmiastowe powiadomienie o nich Zakładu Kolportażu WCT NOT.

Zaznaczamy, że prenumerata ciągła nie dotyczy prenumeratorów indywidualnych, którzy w dalszym ciągu zamawiają czasopisma WCT NOT w urzędach pocztowych do każdego 10, miesiąca poprzedzającego okres prenumeraty — roczny, półroczny, kwartalny.

TADEUSZ OLSZEWSKI

Przemysłowy Instytut Telekomunikacji
Warszawa

681.322.06

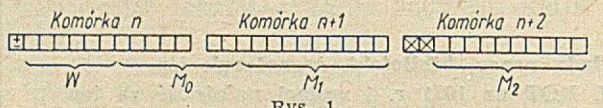
FORPIT — polska modyfikacja 160 FORTRAN-A dla zastosowań metody MONTE-CARLO

W artykule omówiono przyczyny i przebieg opracowania w Ośrodku Obliczeniowym Przemysłowego Instytutu Telekomunikacji (PIT) systemów automatycznego programowania 160 FORTRAN-B i FORPIT. Przedstawiono krótko metodę generacji i testowania liczb pseudolosowych z rozkładem równomiernym i gaussowskim, wytwarzanych za pomocą podprogramów wbudowanych do języka FORPIT oraz eksploatacyjne właściwości tych podprogramów. Zaprezentowano opracowanie metody otrzymywania na EMC liczb pseudolosowych z dowolnym żądanym rozkładem, przeprowadzono przykładowe porównanie wygody eksploatacyjnej i skuteczności zastosowania języków 160 FORTRAN-A i FORPIT do metody MONTE-CARLO.

1. Przyczyny podjęcia w Ośrodku Obliczeniowym PIT opracowania nowego systemu automatycznego programowania 160 FORTRAN-B

Używany w Ośrodku Obliczeniowym PIT do roku 1969 system automatycznego programowania 160 FORTRAN-A nie wykorzystywał możliwości posiadanego zestawu EMC do uzyskania większej efektywnej szybkości liczenia. Równocześnie zgłaszane do ośrodka problemy z zakresu modelowania cyfrowego wymagały posiadania sprawdzonego i dostatecznie wydajnego podprogramu generowania liczb pseudolosowych. Dostawca EMC (firma CONTROL DATA) nie posiada bardziej optymalnej dla ośrodka PIT wersji FORTRANu ani przydatnego praktycznie podprogramu wytwarzania liczb pseudolosowych.

Przyczyny tego można się doszukiwać w samej maszynie CONTROL DATA 160-A, stosunkowo szybkiej (cykl pamięci 6,4 μsek), lecz w całości tylko 12-bitowej stałoprzecinkowej. Opracowany przez producenta maszyny system automatycznego programowania 160 FORTRAN-A zabezpiecza dokładność 8 cyfr znaczących dziesiętnych oraz całkowicie automatyczny programowany zmienny przecinek. Wymaga to programowego łączenia trzech kolejnych komórek pamięci maszyny w jedną 36-bitową komórkę systemową, zmiennoprzecinkową. System automatycznego programowania 160 FORTRAN-A dla maszyny CONTROL DATA 160-A posiada przedstawioną na rys. 1 postać znormalizowanego słowa zmiennoprzecinkowego.



Rys. 1

Najwyższy bit komórki maszynowej n zawiera znak całej liczby, tj. 0 dla liczb dodatnich i 1 dla ujemnych. Części W, M_0, M_1 i M_2 (6, 7, 10 i 10 bitów) przedstawiają dodatnie liczby zapisane w systemie binarnym. Najniższymi bitami są skrajne prawe bity każdej części. Dwa najwyższe bity komórki $n+2$ są nie wykorzystywane. Wartość binarna W przedstawia wykładnik potęgi (przy podstawie 10) plus 32. Ponieważ na 6 bitach części W mogą być zapisane wartości od 0 do $2^6-1 = 63$, to mnożnik 10^{W-32} może zmieniać się w granicach od 10^{-32} do 10^{31}. Części M_0, M_1 i M_2 są częściami mantysy i muszą spełniać warunki:

$$10 \leqslant M_0 < 100, \quad 0 \leqslant M_1 < 1000 \text{ i } 0 \leqslant M_2 < 1000.$$

Z uwzględnieniem powyższych oznaczeń i warunków wartość liczby zmiennoprzecinkowej w postaci znormalizowanej (w jakiej przechowywane są w pamięci EMC) wyraża się w systemie 160 FORTRAN-A wzorem:

$$(1p) \qquad N = \pm 10^{W-32}(M_0 10^{-2} + M_1 10^{-5} + M_2 10^{-8})$$

Wyżej opisany sposób wewnętrznego kodowania liczb zmiennoprzecinkowych jest zapewne wygodny w przypadku języków wyższego rzędu (typu FORTRANU lub ALGOLU) dla maszyn posiadających tylko 12-bitową arytmetykę, ale mniej wygodny dla generacji liczb pseudolosowych. Ponieważ system 160 FORTRAN-A nie pozwala na umieszczanie w programach symbolicznych wstawek napisanych w języku maszyny (lub odpowiedniku SAS), podprogramy wytwarzania liczb pseudolosowych pisano w FORTRANIE. Próbowano zastosować kilka różnych algorytmów generowania liczb pseudolosowych. Żaden z nich nie spełniał podstawowych wymagań. Generowane ciągi liczb pseudolosowych podlegały częstym wyrodzeniom, posiadały za małe okresy powtarzalności i wykazywały niedopuszczalne korelacje między sąsiednimi wartościami. Wydaje się, że przyczyny tych niepowodzeń oraz braku firmowego podprogramu generacji liczb pseudolosowych, odpowiedniego do wykorzystania w programach pisanych w FORTRANIE, należy się doszukiwać w przedstawionym na rys. 1 sposobie wewnętrznego kodowania liczb zmiennoprzecinkowych.

Wymienione wyżej usterki liczb pseudolosowych usunięto przez wprowadzenie dodatkowych komplikacji do podprogramów generacji jak np. przestawianie („mieszanie") w zakresie serii kilkunastu kolejnych liczb, dla rozerwania istniejących więzi korelacyjnych. W rezultacie uzyskano podprogramy dające liczby, z rozkładem równomiernym i gaussowskim, zadawalającej jakości, lecz wymagające znacznej ilości rozkazów w FORTRANIE. Powodowały one już istotne obciążenie pamięci EMC i były zbyt wolne w realizacji.

Każdorazowe użycie tych podprogramów wymagało ponownego częściowego ich „rozgryzania". Niedogodności te nie zachęcały programujących w systemie 160 FORTRAN-A do stosowania metody Monte-Carlo, zwłaszcza programistów niezawodowych.

System automatycznego programowania 160 FORTRAN-A przekształca (przynajmniej dla użytkownika — programisty) maszynę stałoprzecinkową z 12-bitowym słowem i 12-bitową arytmetyką w maszynę operującą na liczbach zmiennoprzecinkowych, z 8-dziesiętnymi cyframi znaczącymi. Jest oczywiste, że wy-

konanie prostych działań zmiennoprzecinkowych, jak np. dodanie zmiennoprzecinkowe dwóch liczb, musi być zrealizowane przez wykonanie w rzeczywistości dość długich programów w języku maszyny (rzędu 100 ÷ 200 rozkazów). Mimo tego, dzięki stosunkowo dużej szybkości wykonywania rozkazów maszynowych, efektywna szybkość realizacji programów napisanych w 160 FORTRAN-A była jeszcze nieco większa od szybkości maszyny ELLIOTT 803 przy programowaniu w MARK 2.

Istniała możliwość istotnego zwiększenia efektywnej szybkości przez wykorzystanie posiadanej w PIT dodatkowej jednostki arytmetycznej 168-2, również produkcji CONTROL DATA. Jest to stałoprzecinkowy arytmometr, wykonujący dodawanie, odejmowanie i mnożenie 27-bitowych liczb całkowitych, dzielenie liczb 54-bitowych przez 27-bitowe, przesuwanie w lewo i prawo oraz przesuwanie ze zliczaniem ilości przesunięć koniecznych do osiągnięcia wartości 1 przez najwyższy (27) bit. Ostatnia właściwość może być wykorzystana do wykonania części operacji normalizacji wyniku działania arytmetycznego. Arytmometr ten był włączony do posiadanego systemu maszyny CONTROL DATA 160-A i mógł być wykorzystywany przy programowaniu w języku maszyny lub systemie SAS ale nie w FORTRANIE. Po prostu producent maszyny nie posiada bardziej odpowiedniej wersji FORTRANU (ani innego języka wyższego rzędu) dla takiego zestawu. Stanie się to bardziej oczywiste, jeśli porównamy przedstawioną na rys. 1 wewnętrzną postać słowa zmiennoprzecinkowego w systemie 160 FORTRAN-A z postacią pożądaną dla optymalnego połączenia 12-bitowej jednostki centralnej z 27-bitowym arytmometrem (Rys. 2).

Wartość liczbowa N klasycznego słowa zmiennoprzecinkowego przedstawionego na rys. 2 wyraża się wzorem:

$$N = \pm 2X\,a \qquad (2p)$$

gdzie $X = b - 128$ oraz $0{,}5 \leqslant a < 1$. Dolna granica a oznacza, że dla znormalizowanej liczby najwyższy bit a (skrajny lewy) musi być jedynką. Przy przewidzianych 8-miu bitach dla zapisu b wartość ta może się zmieniać w granicach od 0 do $2^8 - 1 = 255$. Odpowiada to wartościom X w granicach od —128 do +127. Proste przeliczenie wg wzoru (2p) daje dozwolony zakres absolutnych wartości liczb zmiennoprzecinkowych od 10^{-38} do 10^{38}. Wyniki działań mniejsze od 10^{-38} lub większe od 10^{38} będą automatycznie potraktowane jako maszynowe 0 i maszynowa ∞. Jak widzimy, dodatkową zaletą postaci wewnętrznego kodowania słów zmiennoprzecinkowych przedstawionej na rys. 2 jest rozszerzenie zakresu liczb zmiennoprzecinkowych w stosunku do systemu 160 FORTRAN-A ($10^{-32} \div 10^{31}$).

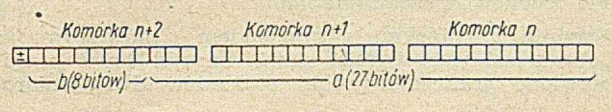
Porównanie obu sposobów kodowania (rys. 1 i rys. 2) wyjaśnia brak odpowiedniej wersji firmowej FORTRANU. Po prostu z powodu zasadniczych różnic między postacią wewnętrznego kodowania w systemie 160 FORTRAN-A i postacią żądaną, przeróbka istniejącego systemu programowania musi być bardzo zasadnicza i pracochłonna (zdaniem niektórych równoznaczna z opracowaniem nowego systemu od początku).

2. Przebieg i rezultat opracowania systemu 160 FORTRAN-B

W Ośrodku Obliczeniowym PIT pracę tę podjął w końcu 1968 r. 2-osobowy zespół bez doświadczenia w tej dziedzinie i zajmujący się równocześnie innymi pracami. Nie posiadano istotnych informacji o budowie i organizacji systemu 160 FORTRAN-A.

Nie znano nawet postaci kodowania przedstawionej na rys. 1. Z konieczności rozpoczęto od najdłuższego

Rys. 2



i najbardziej żmudnego etapu „zdejmowania" dużej ilości funkcjonalnych schematów blokowych (sieci działań) systemu 160 FORTRAN-A na podstawie długich, dosłownie wielotysięcznych sekwencji instrukcji maszynowych. Etap ten mógł być zasadniczo skrócony przez zastosowanie programu automatycznego „zdejmowania" i kreślenia schematów funkcjonalnych typu lansowanego w USA AUTOFLOW. Niestety taki program jest sam dla siebie problemem i wymaga urządzenia pisakowego.

Kolejnym etapem było opracowanie nowych lub poprawienie „zdjętych" schematów funkcjonalnych z uwzględnieniem żądanej postaci wewnętrznego kodowania liczb zmiennoprzecinkowych (Rys. 2). Na podstawie tych schematów pisano w systemie adresów symbolicznych (SAS) poszczególne podprogramy, które SAS zamieniał na ostateczne sekwencje rozkazów w języku maszyny. Tak powstały nowe podprogramy całkowicie automatycznego programowego zmiennego przecinka, standaryzacji, wejścia/wyjścia oraz zmienione podprogramy funkcji wbudowanych do języka, jak SIN, COS, LOG itd. Całość uruchomiono latem 1969 r.

Po przeprowadzeniu dość obszernego programu testowania i usunięciu zauważonych usterek nowy system automatycznego programowania, nazwany **160 FORTRAN-B**, wszedł do normalnej eksploatacji w październiku 1969 r. Zwiększa on 3 do 7 razy szybkość wykonywania działań zmiennoprzecinkowych i obliczania funkcji (w stosunku do systemu 160 FORTRAN-A).

System 160 FORTRAN-B (podobnie jak 160 FORTRAN-A) zamienia maszynę 12-bitową stałoprzecinkową w maszynę 36-bitową z pełnym automatycznym zmiennym przecinkiem. Należy więc do autokodów interpretacyjnych, które charakteryzuje stosunkowo duża ilość operacji organizacyjnych. W naszym przypadku czasy wykonania tych operacji nie uległy istotniejszym zmianom. Dlatego w efekcie uzyskano tylko **3-krotne zwiększenie uśrednionej szybkości** „liczenia" programów pisanych w FORTRANIE.

3. Wykorzystanie nowych możliwości generacji liczb pseudolosowych — system FORPIT

Zrealizowanie w systemie 160 FORTRAN-B postaci kodowania liczb zmiennoprzecinkowych przedstawionej na rys. 2 łącznie z wykorzystaniem dodatkowego arytmometru, pozwala na traktowanie całego systemu maszyny CONTROL DATA 160-A już nie jako maszyny 12-bitowej, lecz 36-bitowej z 27-bitową mantysą i typowym dwójkowym kodowaniem słowa zmiennoprzecinkowego. Można więc zastosować bardziej klasyczne i szybsze sposoby wytwarzania liczb pseudolosowych.

Uwzględniając nowe możliwości oraz „stare" zapotrzebowanie Przemysłowego Instytutu Telekomunikacji na szybki i wygodny dla niezawodowych programistów sposób wytwarzania liczb pseudolosowych, opracowano system FORPIT, będący głównym przedmiotem niniejszego artykułu. Jest on modyfikacją 160 FORTRAN-B, polegającą na wbudowaniu do systemu podprogramów generacji liczb pseudolosowych z rozkładem równomiernym w przedziale (0,1) i liczb gaussowskich normalnych oraz na zastąpieniu słów angielskich polskimi odpowiednikami. Do dotychczasowego repertuaru funkcji jezyka SINF, COSF, ATANF, LOGF, EXPF, ABSF, SIGNF i PRKWF dochodzi jeszcze funkja LLOSF.

Użycie w programie napisanym w FORPICIE wyrażenia LLOSF(0) spowoduje podstawienie na jego wartość kolejnej liczby pseudolosowej z rozkładu równomiernego. Odpowiednio użycie wyrażenia LLOSF(7) spowoduje podstawienie na jego wartość kolejnej liczby z rozkładu gaussowskiego normalnego. Jest to zasadnicze ułatwienie dla programistów, zwłaszcza amatorów, piszących programy oparte na metodzie Monte-Carlo. Zwalnia ono programistę od wszelkich trosk związanych z wykorzystaniem podprogramów liczb pseudolosowych. Stosowanie FORPITU daje rów-

nocześnie istotne zwiększenie szybkości realizacji na EMC w stosunku do dawnych możliwości w systemie 160 FORTRAN-A (6,5 razy dla wytwarzania liczb pseudolosowych i przeszło 4 razy dla pełnych programów modelowania).

FORPIT jest autokodem przystosowanym do zastosowań metody Monte-Carlo, ale zachował całkowicie uniwersalność systemów 160 FORTRAN-A i B. Wbudowany do języka podprogram funkcji LLOSF nie powoduje dodatkowego obciążenia pamięci EMC, jeśli w realizowanym programie liczby losowe nie są używane.

4. Listy zdań języków 160 FORTRAN-B i FORPIT

Poniżej przedstawiono kompletne listy zdań dozwolonych w systemach programowania 160 FORTRAN-B i FORPIT. Zdania na liście języka FORPIT są odpowiednikami zdań umieszczonych w tej samej kolejności na liście języka 160 FORTRAN-B. Wyjątek stanowią dwa ostatnie zdania listy FORPITU, które nie mają odpowiedników na liście 160 FORTRAN-B. Osoby programujące już w języku 160 FORTRAN-B (lub 160 FORTRAN-A), mogą przejść natychmiast do programowania w języku FORPIT bez potrzeby „uczenia się" nowego języka. Programowanie w języku 160 FORTRAN-B, z uwzględnieniem FORPITu, opisano w podręczniku.

Lista zdań języka 160 FORTRAN-B

CALL nazwa (a_1, a_2,...)
CONTINUE
DIMENSION V_1, V_2, V_3,...
Do n I = m_1, m_2, m_3
END
n FORMAT *(sf)*
GO TO *n*
GO TO (n_1, n_2, ... n_m), I
IF wyrażenie n_1, n_2, n_3
IF ACCUMULATOR OVERFLOW n_1, n_2
INPUT lista
NONLOCAL n_1, n_2, ...
OUTPUT lista
PAUSE *n*
PAUSE *n*, I
PUNCH *n*, lista
READ *n*, lista
RETURN
STOP *n*
SUBROUTINE nazwa (a_1, a_2,...)
Z=SINF (X)
Z=COSF (X)
Z=ATANF (X)
Z=EXPF (X)
Z=LOGF (X)
Z=ABSF (X)
Z=SIGNF (X, Y)
Z=SQRTF (X)

Lista zdań języka FORPIT

WYKONAJ nazwa (a_1, a_2,...)
NASTĘPNY
WYMIAR V_1, V_2, V_3,...
DO n I = m_1, m_2, m_3 [1])
KOŃCZ
n FORMAT *(sf)*
SKOCZ DO *n*
SKOCZ DO (n_1, n_2,... n_m), I
GDY wyrażenie n_1, n_2, n_3
GDY NADMIAR n_1, n_2
WEJ BIN lista
WSPÓLNE n_1, n_2,...
WYJ BIN lista
PAUZA *n*
PAUZA *n*, I
DRUKUJ *n*, lista
CZYTAJ *n*, lista
WRACAJ
STOP *n*
PODPROGRAM nazwa (a_1, a_2,...)
Z=SINF (X)
Z=COSF (X)
Z=ATANF (X)
Z=EXPF (X)
Z=LOGF (X)
Z= ABSF (X)
Z=SIGNF (X, Y)
Z=PRKWF (X)
Z = LLOSF (0) rozkład równomierny
Z = LLOSF (7) rozkład gaussowski

5. Podprogram generacji i jakość liczb pseudolosowych

W podprogramie LLOSF liczby pseudolosowe gaussowskie normalne uzyskuje się na podstawie głównego twierdzenia granicznego przez sumowanie n liczb pseudolosowych z rozkładu równomiernego w przedziale (0,1) oraz następującą korekcję wartości średniej i wariancji. Przyjęto n = 7 jako wartość zadowalającą, ale może ona być zmieniona w przypadkach uzasadnionych. Przy wystarczającej wartości n jakość liczb gaussowskich zależy wyłącznie od jakości liczb z rozkładem równomiernym. Dla generowania tych ostatnich wybrano stosunkowo wypróbowaną metodę multiplikatywną, opisaną między innymi w pracy [1] str. 253. W naszym przypadku kolejne wartości pseudolosowe powstają przez realizację algorytmu

$$R_{n+1} = 5^{11} R_n \bmod 2^{27}$$

i posiadają okres $2^{27-2} = 2^{25} = 33\,554\,432$ liczb. Taka wielkość okresu spełnia z dużym zapasem wymagania dla praktycznego stosowania metody Monte-Carlo. Wartość R_0 jest umieszczona w podprogramie na stałe. Zgodnie z [1] może nią być każda liczba dodatnia nieparzysta.

Przeprowadzono testowanie na losowość i zgodność z rozkładem — w stosunkowo szerokim zakresie dla liczb z rozkładem równomiernym i mniejszym dla liczb gaussowskich.

Wykonano następujący zestaw testów, uwzględniający, jak się wydaje, poglądy różnych autorów:

1. wartości średniej (oczekiwanej);
2. częstości *(frequency test)*;
3. częstości rozkładu dwuwymiarowego (test „dwójek");
4. częstości rozkładu trójwymiarowego (test „trójek");
5. par *(serial test)*;
6. kombinacji *(poker test)*;
7. długości serii z podziałem liczb na większe i mniejsze od 0,5;
8. częstości maksimum z 4 kolejnych liczb;
9. częstości minimum z 4 kolejnych liczb;
10. częstości sumy z 7 kolejnych liczb;
11. rozkładu liczb gaussowskich normalnych.

Do porównywania wyników poszczególnych testów z założeniami teoretycznymi stosowano kryterium X^2. Wyniki z wykonania wymienionego zestawu testów (łącznie dla przeszło 15 mln liczb) nie dają żadnych podstaw do odrzucenia hipotezy o zgodności statystycznych parametrów generowanych liczb z parametrami założonymi. W celu niepomnażania już istniejących i stosunkowo licznych opisów (zagranicznych i krajowych) testowania programowych generatorów liczb pseudolosowych, ograniczamy się do informacji, że zwięzły opis metod i pełną dokumentację testowania zawiera opracowanie [8].

Zgodnie z chińskim przysłowiem mówiącym, że „obraz jest wart tysiąca słów", na rys. 3 i 4 przedstawiono graficznie próbki rozkładów dwuwymiaro-

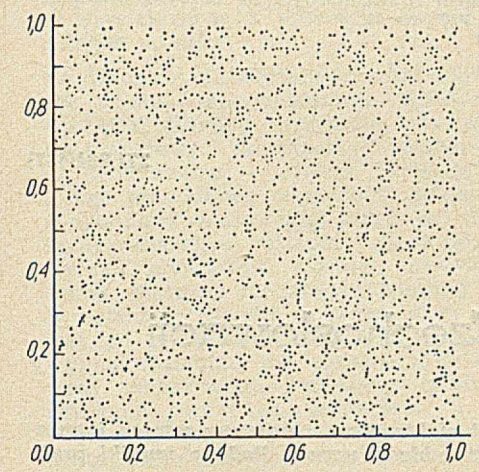
wych, wykonane z dużym nakładem pracy dla rozkładu równomiernego w przedziale (0,1) i gaussowskiego normalnego. Na rysunkach tych, w charakterze współrzędnych kolejnego punktu, przyjmowano wartości dwóch kolejnych liczb pseudolosowych. Przedstawieniom graficznym takiego typu przypisuje się istotną rolę w pracy [2]. Mogą one wykryć istnienie pewnych typów korelacji między liczbami pseudolosowymi (np. układanie się punktów na gęstych liniach równoległych), których nie ujawni zwykły test „dwójek".

6. Zapewnienie elastyczności korzystania z liczb pseudolosowych w systemie FORPIT

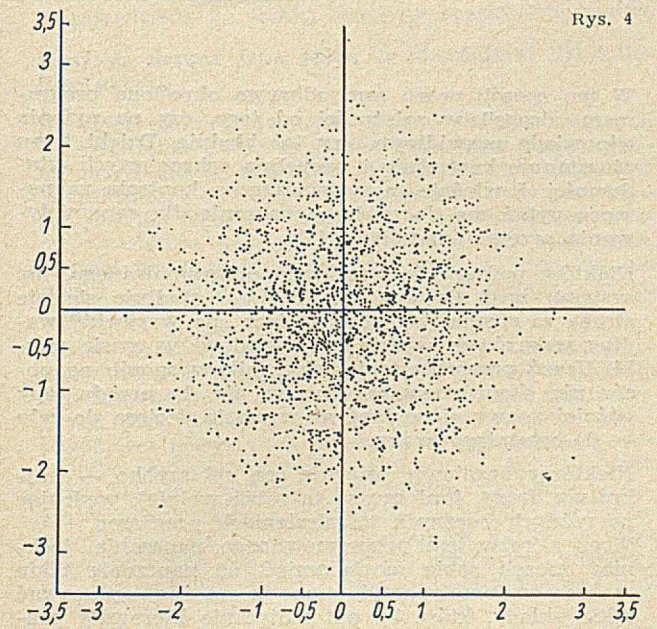
Normalnie każde nowe wczytanie do EMC jakiegokolwiek programu napisanego w FORPICIE i wykorzystującego liczby pseudolosowe powoduje w trakcie jego realizacji generowanie tego samego (powtarzającego się od wartości początkowej R_0) ciągu liczb pseudolosowych. Istnieje możliwość wznowienia realizacji takiego programu po przerwie (w czasie której wykonywano na EMC inne programy), bez powtórzenia ciągu wartości pseudolosowych wykorzystanych przed przerwą. Dla zabezpieczenia tej możliwości należy, najpóźniej przed pierwszym rozkazem zawierającym LLOSF(0) lub LLOSF(7), umieścić rozkaz PAUZA (przeważnie program zawiera już taki rozkaz z innych powodów, np. dla umożliwienia założenia następnej taśmy danych).

Dodatkowo należy zażądać od operatora EMC wyprowadzenia tzw. „mapy pamięci", bezpośrednio po tłumaczeniu programu symbolicznego w FORPICIE.

Rys. 3



Rys. 4



Spełnienie powyższych warunków pozwoli w odpowiednim czasie na bezpośrednie wprowadzenie do pamięci EMC nowej wartości R_0 (dowolna liczba oktalna, dodatnia, nieparzysta i mniejsza od 2^{27}, tj. maksimum 9 cyfr oktalnych).

Dla nieprzekonanych, że każda liczba spełniająca wymienione wymagania może być dobra jako nowa wartość początkowa, zamieszczono kilka wartości sprawdzonych przez wygenerowanie dla każdej z nich 700 tys. liczb z rozkładem równomiernym w przedziale (0,1), np.: 551632437, 327463451, 534512657, 756736415 i 427324323.

7. Generacja liczb pseudolosowych z innymi parametrami

Dla wielu zastosowań metody Monte-Carlo konieczne są liczby pseudolosowe z rozkładem równomiernym w przedziale różnym od (0,1) lub liczby gaussowskie nienormalne. Np. na wartość wyrażenia

LLOSF(0)*C

będą podstawiane kolejne liczby losowe z rozkładem równomiernym w przedziale (0, C). W szczególności przy $C = 2\pi$ kolejne wartości wyrażenia mogą modelować np. losowy równomierny rozkład fazy w przedziale $(0,2\pi)$. Odpowiednio wyrażenie

LLOSF (0)*C—C1

przy $C = 2\pi$ i $C1 = \pi$ może modelować taki rozkład fazy w przedziale $(-\pi, \pi)$. Natomiast wyrażenie

LLOSF (7)*SIGMA+E

będzie dawało w wyniku liczby pseudolosowe gaussowskie z wariancją SIGMA i średnią E.
Przytoczone wyżej przekształcenia liczb pseudolosowych stanowią jedynie najprostsze i oczywiste przykłady. Dysponując liczbami z rozkładem równomiernym i gaussowskim, można stosunkowo łatwo uzyskać większość innych wymaganych rozkładów (w tym np. Rayleigh'a lub Rice'a).

W Ośrodku Obliczeniowym PIT opracowano również metodę praktycznego przekształcenia na EMC liczb pseudolosowych z dowolnym rozkładem uciętym, aproksymowanym liniami łamanymi. Rozkłady te mogą być zadane analitycznie lub graficznie, np. jako wyniki badań eksperymentalnych [10].

8. Przykładowe porównanie przydatności systemów 160 FORTRAN-A i FORPIT do modelowania sygnałów

Przy modelowaniu cyfrowym układów przetwarzania fluktujących sygnałów elektrycznych często spotykamy się z różnymi kombinacjami wyrażeń typu

$$u_i = G \cos(\omega t_i + x)$$

które może przedstawiać np. wartość napięcia w chwili t_i, gdzie G jest niezależną fluktuującą amplitudą z rozkładu gaussowskiego normalnego, a x niezależną fazą losową z rozkładem równomiernym w przedziale $(0,2\pi)$. Pojawienie się takiego wyrażenia w programie napisanym w 160 FORTRAN-A powoduje konieczność umieszczenia w tym programie podprogramów wytwarzania liczb pseudolosowych z rozkładem gaussowskim i równomiernym. Dodatkowo przed każdym rozkazem, wykorzystującym liczby pseudolosowe, konieczne jest umieszczanie rozkazu wywołania odpowiedniego programu. W FORPICIE jest to absolutnie zbędne. Wystarczy napisać:

U(I) = LLOSF(7)*COSF(OMEGA*T(I)+DWAPI*
*LLOSF(0))

Warto zauważyć, że ekwiwalent tego rozkazu w systemie 160 FORTRAN-A wymagał dodatkowo 55 zdań dla dwóch podprogramów liczb pseudolosowych.

*

Powyższy przykład łącznie ze stwierdzeniem, że wbudowane do języka podprogramy liczb pseudolosowych

zajmują kilkakrotnie mniej miejsca w pamięci EMC, niż podprogramy przetłumaczone z FORTRANu, wyjaśniają fakt, że pewne programy oparte na metodzie MONTE-CARLO i napisane w systemie 160-FORTRAN-A do maszyny „nie wchodzą", natomiast mieszczą się w całości w pamięci programy ekwiwalentne napisane w FORPICIE. W sumie pisanie programów opartych na metodzie MONTE-CARLO w języku FORPIT daje:

1. Ułatwienie dla programistów (zwłaszcza amatorów)
2. Skrócenie programów i lepsze wykorzystanie pamięci komputera

3. Zwiększenie szybkości realizacji na komputerze (6,5 dla wytwarzania liczb pseudolosowych i około 4 razy dla pełnych programów modelowania).

Zwiększenie szybkości ma swoje znaczenie, jeśli uwzględnimy fakt, że zgodnie z występującą w PIT specyfiką zagadnień na realizację programów cyfrowego modelowania (należących do metod MONTE-CARLO) zużywano ostatnio około 1000 godzin maszyny cyfrowej rocznie. Wiadomo też, że w świecie komputerów istnieje niepisane prawo, według którego wartość użytkowa i cena komputera jest proporcjonalna do pierwiastka kwadratowego z efektywnej szybkości liczenia.

W miarę zapotrzebowania możliwa jest dalsza specjalizacja FORPITU przez włączanie do języka następnych podprogramów, rozszerzających możliwości stosowania metody MONTE-CARLO (np. podprogramu generacji paczki n skorelowanych liczb według zadawanej kilku punktami funkcji autokorelacji).

Takie podejście jest zgodne z przebijającymi się ostatnio tendencjami do rezygnacji z nadmiernej standaryzacji języków programowania, aprobującymi powstawanie wyspecjalizowanych wariantów językowych dla konkretnych użytkowników [3].

W opracowaniu systemów 160 FORTRAN-B i FORPIT wspólnie z autorem brał udział A. Kuźnicki. Autor wyraża podziękowanie mgr Bożenie Mazurek za przeprowadzenie szerokiego programu testowania, zarówno obu systemów automatycznego programowania, jak i podprogramów generacji liczb pseudolosowych oraz mgr L. Łabanowskiemu za pomoc w rozszyfrowaniu wewnętrznej postaci kodowania słów zmiennoprzecinkowych w systemie 160 FORTRAN A.

[1] Słowo „DO" w wersji polskiej należy rozumieć w kontekście: powtórz poniższą sekwencję rozkazów aż DO rozkazu oznaczonego numerem n włącznie, dla kolejnych wartości $I = m_1, m_1 + m_2, m_1 + 2m_2 + .., m_2$

BIBLIOGRAFIA

[1] A. Ralston, H. S. Wilf: Mathematical Methods for Digital Computers, 1960
[2] R. P. Chambers: Random — number generation on digital computers, IEEE Spectrum 4.2(1967)48.
[3] M. J. Halpern: Toward a General Processor for Programming Languages, Comm. ACM 1/68.
[4] N. P. Buslenko: Metoda Monte-Carlo, tłumacz. z rosyjskiego. PWN 1967.
[5] M. D. Maclaren, G. Marsaglia: Uniform Random Number Generators. J of A.C.M. 12.1(1965)83.
[6] R. Zieliński: Generator liczb losowych o rozkładzie równomiernym dla maszyny ZAM2, Algorytmy 3.6(1966).
[7] A. M. Gadomski, T. Olszewski: Generacja i testowanie liczb pseudolosowych na maszynie cyfrowej CDC 160-A w autokodzie 160 FORTRAN-A; PIT AO B12/65/67 (oprac. wewn.).
[8] T. Olszewski, B. Mazurek: Podprogram generowania liczb pseudolosowych dla języka FORPIT — wybór metody, testowanie generatorów, PIT AO B12/4/70/ (oprac. wewn.).
[9] T. Olszewski, Programowanie maszyny CDC 160-A; język 160 FORTRAN-B, język FORPIT dla zastosowań metody Monte-Carlo, wyd. wewn. PIT 1970 r.
[10] T. Olszewski, Przekształcenie na EMC liczb pseudolosowych z rozkładem równomiernym na liczby z rozkładem zadanym, Prace PIT, nr 72.

ZDENEK KŘEČAN
Praha — CSRS

371.694(437)

Maszyny uczące w Czechosłowacji

Przedstawiono opracowane w Czechosłowacji elektroniczne maszyny uczące i egzaminujące oraz maszyny rozróżniające charakter błędu ucznia. Podano wyniki praktycznego zastosowania tych urządzeń.

Żyjemy w okresie szybkiego rozwoju nauki, nowych odkryć i wynalazków. Człowiek musi przyswajać sobie coraz to wiekszą ilość wiadomości. W związku z tym w szkołach zaczęto stosować rozmaite poglądowe pomoce naukowe, od urzadzeń najprostszych aż po skomplikowane takie, jak film dźwiękowy, radio i telewizja.

Wraz z rozwojem środków masowego przekazu, w ostatnim czasie pedagodzy skoncentrowali uwagę na nauczaniu programowanym. Od nauczania tradycyjnego różni się ono głównie tym, że materiał dydaktyczny zostaje tu podzielony na małe zwarte fragmenty-informacje, zwane **krokami.** Oczywiście kolejność tych kroków musi być logiczna. Bezpośrednio po każdym kroku nastepuje pytanie kontrolne. Uczeń na to pytanie odpowiada w ten sposób, że albo wpisuje odpowiedź w odpowiednim miejscu specjalnego podręcznika, albo też z kilku wariantów odpowiedzi do wyboru wybiera tę, którą uważa za prawidłową. Następnie koryguje on swoje odpowiedzi, porównując je z rozwiązaniami podanymi w innym miejscu podręcznika.

W ten sposób uczeń sam odkrywa określoną programem „drogę", w zależności od tego, czy na pytania odpowiada prawidłowo, czy też błędnie. Dzięki temu nieustannie kontroluje i uzupełnia zakres swych wiadomości. I właśnie ta systematyczna kontrola za pomocą pytań ma decydujące znaczenie dla zapamiętywania przerabianego materiału.

Niektóre podręczniki nauczania programowanego, np. systemu prof. Crowdera z USA — studiuje się nie strona za stroną, jak książkę, lecz „na wyrywki" według wskazówek udzielanych uczniowi w czasie nauki. Uczeń przerabia taki podręcznik w sposób na pozór nie skoordynowany, posuwa się do przodu, aby później znowu sie cofnąć aż wreszcie dociera do celu — do ostatniego **kroku.**

Niektórzy uczniowie osiągają ten cel szybko — najkrótszą drogą, inni muszą to robić wolniej, wędrując po różnych bocznych odgałęzieniach programu.
Wraz z rozwojem programowanego nauczania, technicy zaczęli sobie uświadamiać, że nauczanie takie można zautomatyzować oraz że można opracować takie układy, które by automatycznie kierowały pra-

cą ucznia i regulowały napływ informacji. Powstały w ten sposób maszyny uczące.

Maszyny uczące

Maszyna ucząca jest w gruncie rzeczy zautomatyzowanym podręcznikiem, w którym odpowiedzi ucznia są poddawane obiektywnej ocenie. Wraz z rozwojem techniki, a zwłaszcza elektroniki — konstrukcja maszyn uczących uległa zasadniczym przeobrażeniom. Nowoczesne maszyny uczące mają konstrukcję całkowicie elektroniczną. Nie wyeliminowały one jednak prostych maszyn typu mechanicznego, których zaletą jest bardzo niski koszt.

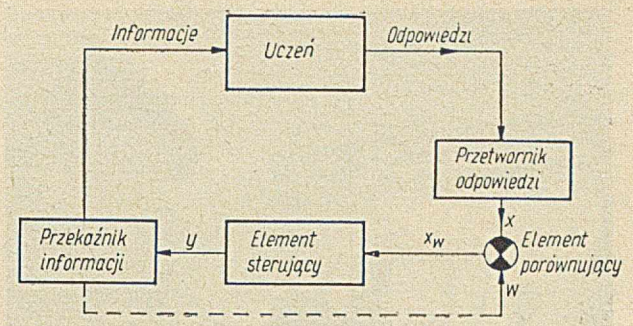
Wydzieloną grupę maszyn uczących tworzą tzw. egzaminatory i repetytory, które służą do sprawdzania wiadomości i do prostego powtarzania materiału. W Czechosłowacji produkuje się dwa typy tego rodzaju maszyn. **„Zkušaci stroj"** (Maszyna egzaminująca) jest produktem przedsiębiorstwa państwowego TESLA-Vrable. Jest to całkowicie zautomatyzowane urządzenie do egzaminowania uczniów, a także do prostego powtarzania materiału. Natomiast maszyna ZUZA wyprodukowana przez przedsiębiorstwo wojskowe, pomimo stosunkowo prostej konstrukcji znajduje bardzo szerokie zastosowanie. Uczeń otrzymuje zeszyt z programem, który kładzie na górną płytę maszyny. Na pytania odpowiada wciskaniem przycisków. Maszyna światłem sygnalizuje uczniowi popełnione błędy, które równocześnie notuje na liczniku. Zestaw maszyn uczących ZUZA może także służyć do nauczania całej grupy uczniów równocześnie. Każdy uczeń w klasie pracuje na jednej maszynie, która jest połączona kablem z centralnym urządzeniem kontrolnym u nauczyciela. Na tablicy świetlnej nauczyciel może z łatwością śledzić pracę poszczególnych uczniów.

Na wydziale elektrotechnicznym politechniki w Pradze działa eksperymentalnie AUTOMAT EGZAMINUJĄCY. Student otrzymuje trzy pytania napisane na papierowych kartkach. W dolnej części, karty te są perforowane — otwory określają numer pytania i jego prawidłową odpowiedź. Student wkłada karty z pytaniami w dowolnej kolejności do szufladki maszyny uczącej, a następnie odpowiada na pytania za pomocą przycisków, stosując trzycyfrowy kod dziesiętny. Maszyna na tablicy świetlnej w sposób trwały wskazuje wyniki poszczególnych odpowiedzi. Nawet po wyłączeniu urządzenia z sieci elektrycznej wyniki egzaminu nie zostaną zatarte, ponieważ obwody elektroniczne maszyny pamiętają wszystkie uzyskane poprzednio odpowiedzi i po ponownym włączeniu urządzenia wszystkie popełnione przez studenta błędy natychmiast zostaną wykazane. Tylko nauczyciel może wymazać rezultaty egzaminu przez wprowadzenie do zamka maszyny specjalnego klucza.

Maszyna ucząca jako jeden ze składników zamkniętego układu

Bardziej skomplikowanym typem maszyn uczących są urządzenia, które w sposób całkowicie automatyczny przedkładają uczniowi zaprogramowany materiał. W zależności od tego, czy odpowiedź była prawidłowa, czy też błędna — maszyna w sposób automatyczny wyszukuje kolejne optymalne kroki nauczania. Najczęściej jednak stosuje się metodę wyświetlania tekstu z przezroczy przy użyciu automatycznego rzutnika.

Zasady działania polegają na tym, że na ekranie maszyny uczącej wyświetla się jedną pozycję materiału dydaktycznego, po czym następuje pytanie kontrolne. Uczeń odpowiada za pomocą przycisków. Jeśli odpowiedź jest prawidłowa, maszyna to potwierdza i wyświetla następną pozycję materiału — kolejny krok. Jeśli jednak odpowiedź jest błędna, projektor automatycznie wyszukuje przezrocze z bardziej szczegółowym objaśnieniem i skorygowaniem błędu, a więc maszyna stara się naprowadzić ucznia na właściwy tor i pomóc mu w zrozumieniu podstawowego materiału.



Rys. 1. Obwód zamknięty maszyny uczącej

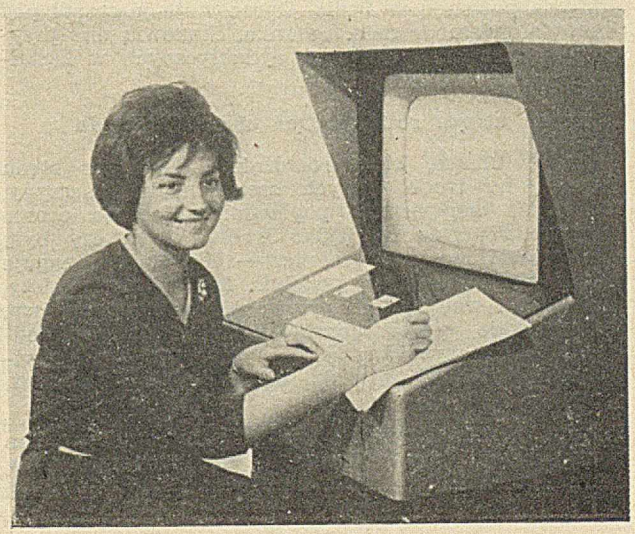
Na taką maszynę uczącą możemy patrzeć, jak na urządzenie techniczne, które stanowi część składową zamkniętego obwodu, przedstawionego na rys. 1. Kierowanym obiektem jest tu uczeń. Jego stan czyli zasób jego wiadomości stwierdza się za pomocą **przetwornika odpowiedzi.** Może to być np. zestaw przycisków, tarcza telefoniczna lub inne urządzenie, którego uczeń używa do przetransponowania swojej odpowiedzi na język maszyny.

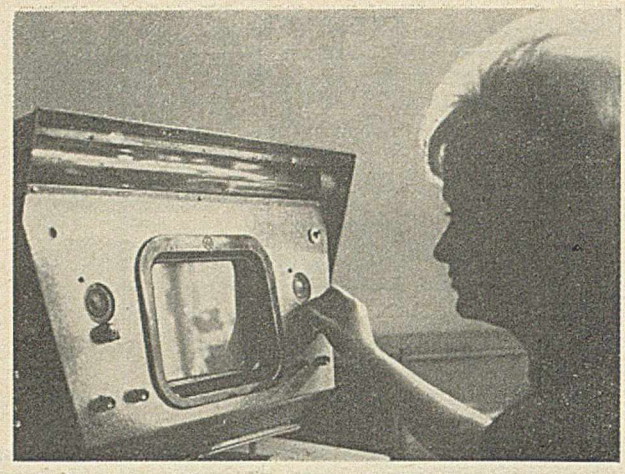
Element porównujący porównuje odpowiedź ucznia (w postaci wielkości x) z żądaną wielkością w. Rezultatem tego porównania jest sygnał x_w doprowadzany do **elementu sterującego.**

Element sterujący przesyła sygnały mechaniczne do **przekaźnika informacji,** który z kolei pośredniczy w przekazaniu uczniowi nowych wiadomości z przerabianego materiału, poprawia jego błędy i stawia pytania. Może to być np. rzutnik z automatyczną wymianą przezroczy, magnetofon, lampa oscyloskopowa itp. Tak więc element sterujący przesyła np. do automatycznego rzutnika sygnały (y) nakazujące zmianę potrzebnej liczby przezroczy w kierunku do przodu lub do tyłu. Nieodłączną częścią przekaźnika informacji jest program nauczania, który oprócz informacji przeznaczonych dla ucznia musi zawierać dane definiujące sygnał — w, czyli informację o prawidłowym brzmieniu odpowiedzi.

W Czechosłowacji przed kilkoma laty opracowano pierwszy prototyp maszyny uczącej KT 1 (patrz rys. 2), oparty na zasadzie projekcji przezroczy. Sygnał w otrzymywano tu z obrzeża przezrocza, na którym prawidłowa odpowiedź była zakodowana w układzie dwójkowym w postaci czarnych i białych pól świetlnych. Oczywiście sygnał ten nie był wyświetlany na ekranie, lecz kierowano go na układ fotokomórek

Rys. 2. Maszyna ucząca KT-1





Rys. 3. Maszyna ucząca dla kształcenia kadr służby zdrowia

umieszczonych na skraju matówki projekcyjnej. Uzyskane sygnały fotoelektryczne za pomocą urządzenia dekodującego zostają zmienione na jedno-, dwu- lub trzycyfrową liczbę w układzie dziesiętnym, a obwody logiczne porównują je z odpowiedzią ucznia. W zależności od odpowiedzi urządzenie automatycznie przygotowuje kolejne przezrocze.

Z prototypu KT 1 powstała obecna maszyna ucząca KT 1-X EVA. Jej częścią składową jest m.in. produkowany w CSRS automatyczny rzutnik ADIOR. Odpowiedzi przenosi uczeń na maszynę za pomocą jednego z pięciu przycisków. Istnieje również nowa wersja tej maszyny o nazwie KT 2; jest ona wyposażona w 10 przycisków, a kod ma postać trzycyfrowej liczby dziesiętnej. Kolejny model maszyny uczącej KT 3, który znajduje się obecnie w eksploatacji próbnej, ma również odpowiedzi zakodowane w formie trzycyfrowych liczb dziesiętnych, lecz jest wyposażony w nowy specjalny rzutnik MEOPTA-AUTODIOR oraz realizuje bardziej rozgałęziony program w kierunku do przodu i do tyłu.

Warto jeszcze wspomnieć o maszynie uczącej typu KT 4, która działa na podobnych zasadach, lecz wyświetlanemu obrazowi towarzyszy komentarz z magnetofonu. Maszyna ta jest przeznaczona do nauczania całej grupy uczniów równocześnie. Każdy uczeń ma przed sobą małą tablicę, na której przełącznikiem nastawia swą odpowiedź. Maszyna podlicza odpowiedzi błędne oraz prawidłowe i w zależności od ogólnego rezultatu albo wprowadza następny krok, albo też materiał powtarza.

Interesujące i pomysłowe maszyny uczące, przeznaczone przede wszystkim do kształcenia kadr służby zdrowia powstają w Brnie. Opracowuje je zespół pod kierownictwem dr med. Šatanka. Stanowią one przykład zastosowania urządzeń dydaktycznych w kształceniu pozaszkolnym (rys. 3).

Maszyny uczące rozróżniające charakter błędu

Maszyny uczące, które rozróżniają charakter błędu działają na trochę innych zasadach, niż maszyny, o których wspomniano poprzednio. Maszyna ocenia odpowiedź ucznia nie tylko według tego, czy była ona prawidłowa czy też nie, — lecz także reaguje na to, **jaki** błąd uczeń popełnił. Ma to swoje zalety, ale także i wady, które przeważnie polegają na tym, że na każde pytanie uczniowie otrzymują kilka odpowiedzi **do wyboru**, tzn. że uczeń wybiera prawidłową odpowiedź spośród kilku innych nieprawidłowych.

Zasadę tę można najlepiej wyjaśnić na prostym przykładzie. Maszyna ucząca jest zaopatrzona w cztery przyciski do udzielania odpowiedzi — A, B, C i D.

Jeśli uczeń wciśnie przycisk A, automat przesuwa w rzutniku pojemnik z przezroczami o jedną klatkę do przodu. Przyciskiem B następuje przesunięcie o dwa przezrocza, przyciskiem C — o trzy przezrocza, a przyciskiem D — o cztery przezrocza do przodu.

Np., gdy zaprogramowane pytanie brzmi: ile to jest 9^2? — Uczeń ma do wyboru cztery następujące warianty odpowiedzi:

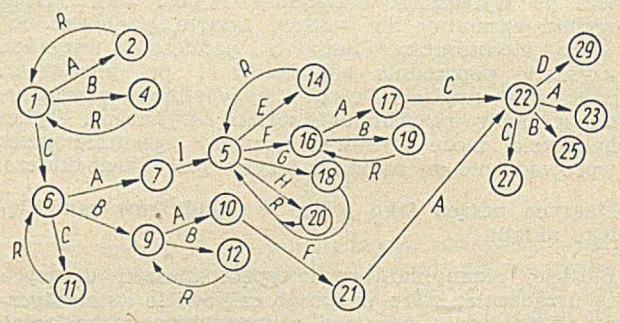
$9^2 = 18$	wciśnij A
$9^2 = 3$	wciśnij B
$9^2 = 81$	wciśnij C
nie wiem	wciśnij D

Po wciśnięciu jednego z przycisków uczeń otrzymuje na ekranie prawidłową odpowiedź, a równocześnie wyjaśnienie, na czym polegał jego błąd lub też potwierdzenie, że jego odpowiedź była błędna. W tym ostatnim przypadku zostaje wzbudzony sygnał fotoelektryczny, który powoduje przesunięcie zasobnika z przezroczami na kolejny temat. Natomiast w razie popełnienia błędu po ponownym wciśnięciu przycisku rzutnik podaje jeszcze raz poprzednie przezrocze, aby uczeń mógł powtórzyć sobie dany materiał.

Na tej właśnie zasadzie działa najbardziej znana dziś projekcyjna maszyna ucząca AUTOTUTOR MARK II. Jest ona produkowana seryjnie w USA i opracowano dla niej bardzo dużo różnych programów. Jest wyposażona w 10 przycisków, w tym 9 przycisków do udzielania odpowiedzi, które są oznaczone literami od A do I. Dziesiątym jest przycisk R. Poszczególne przyciski przesuwają film z tekstem w następujący sposób:

przycisk A o jedną klatkę do przodu
przycisk B o 3 klatki do przodu
przycisk C o 5 klatek do przodu
aż do H o 15 klatek do przodu.

Przycisk I spełnia odmienną rolę, przesuwa film o trzy klatki do tyłu. Za pomocą przycisku R uczeń automatycznie cofa się na początek ostatniego kroku. Wymieniony tu system stwarza możliwość licznych i różnorodnych rozgałęzień programu i kombinacji.



Rys. 4. Struktura jednego kroku programu, opracowanego dla maszyny uczącej AUTOTUTOR MARK II

Na rys. 4 jest pokazana struktura jednego kroku programu opracowanego dla maszyny uczącej AUTOTUTOR-MARK II.

W CSRS maszynę AUTOTUTOR-MARK II poddano dokładnym badaniom, a jej działanie zostało wymodelowane na prostej maszynie matematycznej. Dzięki temu można było trochę zmodyfikować jej funkcje i opracować optymalny system kombinacji i rozgałęzień programu. Z tego eksperymentu powstała czechosłowacka maszyna ucząca TRANZI-TUTOR II, która jest znacznie prostsza niż maszyna amerykańska i wydaje się, że także jej koszt produkcji będzie stanowił zaledwie ułamek ceny maszyny zagranicznej.

Praktyczny eksperyment w ramach szkolenia pozaszkolnego

W Czechosłowacji maszyny uczące znajdują praktycznie zastosowanie w wielu dziedzinach — w wojsku, przy szkoleniu kierowców, przy dokształcaniu pracowników służby zdrowia, w szkołach itp. Największy eksperyment w dziedzinie kształcenia za pomocą maszyn uczących zorganizowała w Pradze SOCIALISTICKA AKADEMIE (odpowiednik polskiego TWP — przyp. tłum.) organizacja, której jednym z zadań jest dokształcanie pozaszkolne.

Przy współpracy z Wydziałem Pedagogicznym Uniwersytetu Karola w Pradze SOCIALISTICKA AKADEMIE zorganizowała w Październiku 1967 r. gabinet wyposażony w 20 maszyn uczących. Wśród nich jest 5 maszyn typu EVA, 13 maszyn typu HANKA oraz 3 maszyny typu ZUZA. Różne typy maszyn stosuje się po to, aby można było praktycznie wypróbować i porównać poszczególne rozwiązania.

We wspomnianym gabinecie jest prowadzony zautomatyzowany jednosemestrowy kurs „Półprzewodniki w elektrotechnice". Na kursie tym słuchacze zapoznają się z podstawami techniki półprzewodników oraz graficznego przedstawiania układów tranzystorowych.

Kurs jest podzielony na 10 dwugodzinnych lekcji, zaś każda lekcja wymaga średnio 2 do 3 godzin nauki. W chwili obecnej z takiego zautomatyzowanego kursu korzysta 140 pracowników, oddelegowanych tu przez różne zakłady pracy.

W czasie trwania kursu, pracownicy naukowi Wydziału Pedagogicznego prowadzą swe prace badawcze. Obserwują oni słuchaczy za pomocą telewizji przemysłowej ze swego laboratorium, gdzie też są rejestrowane przez maszynę matematyczną wszystkie odpowiedzi uczniów.

W najbliższym czasie w tym eksperymentalnym ośrodku organizacji SOCIALISTICKA AKADEMIE, rozpoczną się kursy chemii i matematyki dla kandydatów przygotowujących się do egzaminów wstępnych na wyższe uczelnie.

Czy maszyna ucząca może zastąpić nauczyciela?

Maszyny uczące powinny przede wszystkim uwolnić specjalistów-pedagogów od takich wyczerpujących i nużących prac, jak np. poprawianie prac pisemnych, egzaminowanie, powtarzanie materiału dydaktycznego itp.

Maszyny przeznaczone do nauczania zbiorowego już w trakcie samej nauki mogą oceniać aktualne wiadomości całej grupy uczniów, a nauczyciel na tej podstawie może na bieżąco korygować swój wykład, zmieniać jego tempo, a w razie potrzeby peszcze raz powtórzyć dany materiał.

Istnieje cały szereg przykładów świadczących o korzyściach, jakie płyną ze stosowania maszyn uczących.

Angielski profesor Henry Mc Kay po przeprowadzeniu rozległych badań doszedł do wniosku, że dzięki zastosowaniu maszyn uczących przy nauczaniu określonych zagadnień z geometrii wykreślnej — liczba godzin niezbędnych do opanowania całości materiału zmniejszyła się z dziesięciu do czterech i pół. Okazało się przy tym, że studenci nauczani za pomocą maszyn uczących po miesiącu lepiej pamiętali materiał dydaktyczny, niż uczniowie, którzy uczyli się sposobem „klasycznym". Ale maszyny mają także wielu przeciwników. Ogólnie przyjęty jest pogląd, że głównym zadaniem nauczania w szkołach jest rozwijanie samodzielnego myślenia. Uczeń w pierwszym rzędzie musi opanować umiejętność uczenia się. A czy maszyny uczące rozwijają tego rodzaju zdolności?

Na takie i tym podobne pytania nauka dziś jeszcze nie może dać definitywnej odpowiedzi. Taką odpowiedź da nam dopiero praktyka. Jest jednak rzeczą bezsporną, że programowane nauczanie i maszyny uczące będą szeroko stosowane, zarówno w szkołach, jak też i w kształceniu pozaszkolnym. Potwierdzeniem tej tendencji jest m.in. także wyżej wspomniany eksperyment SOCIALISTICKA AKADEMIE w Pradze, choć kurs poświęcony półprzewodnikom jest dopiero pierwszym krokiem na tym polu.

Oczywiście maszyny uczące nigdy nie zdołają całkowicie zastąpić nauczyciela, lecz mogą stać się jego pożytecznym pomocnikiem i współpracownikiem, który wydatnie ułatwi jego męczącą i odpowiedzialną pracę.

ZE ŚWIATA

II MIĘDZYNARODOWA KONFERENCJA SZTUCZNEJ INTELIGENCJI

W aulach wykładowych Imperial College w Londynie odbędzie się w dniach 1—3.IX.1971, tj. w tygodniu bezpośrednio następującym po Kongresie IFIP w Jugosławii, impreza naukowa pt. „International Joint Conference on Artifical Intelligence". Bezpośrednim organizatorem tej Konferencji jest Brytyjskie Towarzystwo Komputerowe, działające z ramienia Międzynarodowej Połączonej Rady do Spraw Sztucznej Inteligencji, powołanej w wyniku ustaleń pierwszej takiej konferencji odbytej swego czasu w Waszyngtonie.

Organizatorzy przyjmują zgłoszenia z następujących dziedzin:

- Podstawy teoretyczne sztucznej inteligencji
- Badania eksperymentalne sztucznej inteligencji
- Dowodzenie twierdzeń
- Heurystyczne rozwiązywanie problemów
- Nauczanie maszyn
- Percepcja kształtu
- Rozgrywanie gier
- Systemy interakcyjne
- Sztuczna inteligencja w aspekcie lingwistyki
- Realizacja systemów ze sztuczną inteligencją
- Środki techniczne i oprogramowanie sztucznej inteligencji
- Współpraca człowiek-maszyna w rozwiązywaniu problemów
- Modelowanie psycho- i fizjologiczne
- Zastosowanie teorii systemów ze sztuczną inteligencją.

Zgłoszenia propozycji nadesłania referatu należało kierować w nieprzekraczalnym terminie do dnia 1 listopada 1970 roku pod adresem: Prof. D. C. Cooper, Computer Science Dept. University College of Swansea. Singleton Park, Swansen, Glamonganshire, załączając 500-wyrazowe streszczenie (w języku angielskim) z prowizorycznym tytułem referatu, opatrzone pełnymi imionami i nazwiskiem, dokładnym adresem oraz określeniem instytucji zatrudniającej i zajmowanej pozycji (należy dołączyć ewentualną zgodę kierownika instytucji zatrudniającej na nadesłanie referatu omawiającego badania naukowe wykonywane na określone zlecenie).

Po zakwalifikowaniu wstępnym udziału w programie naukowym, należy w nieprzekraczalnym terminie 28 lutego 1971 roku nadesłać kompletne maszynopisy referatów, według ściśle określonych wymagań, które otrzymają w stosownym czasie wszystkie zakwalifikowane osoby.

A.B.E.

Problemy szkolenia programistów w USA[1)]

Przytoczono analizę organizacji, metod i kosztów szkolenia programistów na kursach.

Zagadnienie szkolenia programistów dla potrzeb rozwijającej się dynamicznie komputeryzacji naszego życia i działalności wszystkich niemalże dziedzin gospodarki staje się zagadnieniem problemowym nr 1. Szczególnie interesujące są elementy charakteryzujące nakłady związane ze szkoleniem, a z drugiej strony korzyści tego szkolenia, wyrażone efektywniejszą pracą przeszkolonych programistów. Problemem tym zajmuje się wielu specjalistów całego świata.

Niniejszy artykuł opracowano na podstawie publikacji Stanley'a L. Sawyera[1)], starszego programisty centrum APD w General Dynamics, specjalizującego się w zagadnieniach szkoleniowych. Autor próbuje dokonać analizy kosztów szkolenia, przedstawia pewne formuły matematyczne z tym związane oraz wyciąga wnioski ze swoich rozważań.

Czy zmiana w polityce kosztów edukacji programistów wywoła negatywne skutki w postaci niedostatecznych kwalifikacji? Czy programiści, szczególnie ci, którzy będą związani z problemami technicznymi, sprostają postawionym przed nimi zadaniom? Pytania tego typu nurtują szeroki krąg użytkowników komputerów na całym świecie, w tym również w Stanach Zjednoczonych AP, gdzie obecnie zainstalowanych jest około 80 000 komputerów, a liczba ich zwiększa się rocznie o 15—20%. Wartość pracujących już komputerów wynosi około 20 miliardów dolarów. Roczne koszty personelu utrzymującego instalacje w ruchu kształtują się na poziomie prawie 80% wartości zainstalowanych urządzeń.

Wyraźnie więc widać, że koszt instalacji w dużym stopniu jest uzależniony od nakładów na utrzymanie jej w stanie zdolności produkcyjnej. A więc im personel jest lepiej przeszkolony, tym bardziej jest efektywny i przyczynia się do zmniejszenia kosztów. Według ostatnich ustaleń jednego z naukowców amerykańskich — dra R. M. Fumlera z Georgia State School of Business, niektóre koncerny mogłyby w krótkim czasie wyasygnować więcej środków na cele szkoleniowe niż oficjalne placówki dydaktyczne.

Przykładowo budżety szkoleniowe firm IBM i GENERAL ELECTRIC są większe od budżetów większości uniwersytetów amerykańskich. Ponieważ szkoły nie nadążają ze szkoleniem kadr dla ośrodków obliczeniowych, przemysł nie ma wyboru i musi sam szkolić potrzebnych specjalistów. Z tego powodu kierownictwo ośrodków APD coraz większą wagę przykłada do jakości i kosztu szkolenia. Obecnie w USA brak jest prawie 60 000 programistów. W 1968 r. było ich około 120 000, a w roku 1970 zapotrzebowanie wzrosło do około 250 000. Według przewidywań wiceprezydenta koncernu GENERAL ELECTRIC — J. S. Smitha, w 1973 r. potrzeba będzie aż 500 000 programistów. Oznacza to, że aby sprostać temu zapotrzebowaniu, należy w przeciągu 4 lat wyszkolić armię 300 000 ludzi.

Zagadnienie to przeobraża się w poważny problem z uwagi na fakt rozproszenia wysiłków w zakresie szkoleniowym, które ostatnio ostro się zarysowało. Kto więc w rezultacie ma zająć się szkoleniem tak licznej kadry? Czy użytkownicy komputerów we własnym zakresie? Czy wobec otwierania coraz to nowych szkół dla programistów istniejące ośrodki szkoleniowe u producentów komputerów będą miały rację bytu?

W coraz większej ilości szkół i uniwersytetów uzupełnia się programy o kursy z zakresu przetwarzania danych. Możliwe więc, że w dalszej perspektywie łatwiej będzie o programistę z ukończonym szkoleniem specjalistycznym, ale co będzie w najbliższej przyszłości?

W celu zbadania opinii szerszego kręgu zainteresowanych przeprowadzono w październiku 1969 r. ankietyzację wśród stu członków organizacji GUIDE. Ankieta zawierała pytania związane z obecnymi metodami szkolenia oraz efektami oczekiwanymi w macierzystych ośrodkach respondentów. Większość z czterdziestu instytucji, które nadesłały odpowiedzi, stwierdziła, że system szkolenia dostosowany do potrzeb poszczególnych odbiorców może przynieść pozytywne rezultaty. Jednakże w sporej części odpowiedzi podkreślano, że jakość szkolenia jest niedostateczna. Znamienne, że zarzuty kierowano pod adresem tak renomowanej potęgi, jaką jest IBM. Niezadowolenie z jakości szkolenia oferowanego przez tę firmę okazało się bardzo duże. Jeżeli w przyszłości IBM nie zmieni tej sytuacji na lepsze, to kontrahenci przestaną korzystać ze szkolenia i nie będą za nie płacić.
Powyższa dezaprobata respondentów ma swoje podłoże w złej organizacji, nieodpowiednim personelu i niewłaściwym ustawieniu czasu nauki. Słuchacze reprezentują krańcowo różne poziomy zaawansowania — od zupełnie surowego materiału do kilkuletniej praktyki. Taki układ sił nie gwarantuje pozytywnych rezultatów szkolenia w jednej grupie słuchaczy.

Efektywność szkolenia można poprawić przez staranny dobór kadydatów na programistów i zaprzestanie praktyki niemalże automatycznego typowania. Z drugiej strony musi istnieć możliwość porównywania kursów prowadzonych przez rozmaite instytucje pod względem jakości, programu i kosztu. Celem tego porównania byłoby uzyskanie jak najlepszych rezultatów szkolenia przy możliwie najniższych nakładach drogą wyboru odpowiedniej szkoły. Wielu poważniejszych użytkowników komputerów, rozczarowanych poziomem szkolenia u producentów sprzętu, zorganizowało kursy we własnym zakresie. Liczny odsetek badanych przedsiębiorstw, mających niewielkie lub żadne możliwości szkolenia na własnym terenie, organizuje obecnie szkolenie w celu obniżenia kosztu edukacji. Są to jednak tylko te placówki, które mogą zapewnić wystarczającą kadrę wykładowców. Rozwiązanie to ma pewne zalety, bo jest bardziej opłacalne finansowo i pozwala uzyskać programistów przeszkolonych według własnych potrzeb organizacyjnych.

Metody szkoleniowe

Większość z rozpatrywanych powyżej użytkowników komputerów prowadzi bądź dopiero organizuje kursy szkoleniowe podstaw programowania. Kursy tego typu są również oferowane przez producentów komputerów. Większość z nich charakteryzuje dobra jakość i mogą być łatwo dostosowywane do potrzeb

---

[1)] Sawyer S. L.: Unbundling's Impact on Training of Programmers, Journal of Data Management, marzec 1970.

konkretnego odbiorcy. Do programu takiego kursu włączona jest również praktyka, a zatem słuchacze oprócz wiedzy podstawowej po ukończonym kursie mogą także legitymować się pewnym doświadczeniem. Coraz szerzej wprowadza się nowoczesne metody szkoleniowe z użyciem wielu pomocy naukowych w postaci projektorów, rzutników, tablic poglądowych itp. Jednakże mimo dostępnego szerokiego wachlarza filmów szkoleniowych zainteresowanie nimi jest raczej niewielkie, jakkolwiek udowodniono, że ten rodzaj przekazu jest wysoce efektywny.
Większe firmy sięgają po inne jeszcze środki, jak np. magnetowidy. Jest to szczególnie korzystne wówczas, gdy nie ma możliwości zapewnienia wysoko kwalifikowanej kadry instruktorskiej. Największe problemy stwarza szkolenie projektantów systemów. Brak wykładowców o odpowiednio dużej wiedzy technicznej zmusza do wysyłania ludzi na kursy prowadzone przez producentów komputerów.
Koszt takiego przeszkolenia 1 osoby w ośrodku IBM wynosi około 1500 dol.

Możliwości szkolenia

Okazuje się, że w najgorszej sytuacji znajdują sie niewielkie placówki. Z reguły nie mogą pozwolić sobie na prowadzenie własnych kursów z braku wykwalifikowanej kadry instruktorskiej. Pozostaje im przeto powierzanie edukacji komuś z zewnątrz, a to pociąga za sobą wzrost kosztów. Dlatego zaleca się tym użytkownikom korzystanie z kursów o charakterze podstawowym jako drogę do uzyskania niezbędnych kwalifikacji mniejszym kosztem.
Z tego też względu w przyszłości będzie się kłaść nacisk na to, aby dla małych przedsiębiorstw ograniczyć zakres szkolenia.
Zgodnie ze wspomnianą wyżej ankietą prawie 64% programistów zatrudnionych w niewielkich ośrodkach legitymuje się ponad roczną praktyką. Z drugiej strony ośrodki, korzystające z komputerów na zasadzie dzierżawy z miesięczną opłatą ponad 500 000 dolarów, zatrudniają około 65% programistów nie mających żadnej praktyki. Wniosek, jaki sie tu nasuwa, dowodzi przechodzenia wyszkolonych ludzi z ośrodków większych do mniejszych. Większość respondowanych przedsiebiorstw wykazała dążność do ograniczenia szkolenia poza własnym terenem na korzyść kursów organizowanych u siebie. Według opinii dyrektora centrum szkoleniowego IBM trend ten będzie sie utrzymywał do początku roku 1970, potem zaś pod koniec 1971 r. nastąpi prawdopodobnie powrót do stanu normalnego.

Koszt szkolenia

Wysokość wydatków na szkolenie zależna jest od programu szkolenia. Członek grupy GUIDE — Turner Wilson w listopadzie 1969 przedstawił następująca formułę matematyczną, pozwalającą wyliczyć roczny koszt szkolenia programistów:

$$E_c = ((C_t (T + G)(A + S))) + (C_1 (T_1 (P_1 + A))) + (C_s (T_s S))$$

gdzie:

E_c — koszt szkolenia.

C_t — koszt nauki programowania,

T — procent ubytku kadr rocznie,

G — procent rocznego wzrostu zatrudnienia,

A — średnia liczba programistów,

S — średnia liczba projektantów systemów,

C_1 — koszt kursu dla programistów prowadzących,

T_1 — procent ubytku rocznego programistów prowadzących,

P_1 — procent programistów prowadzących,

C_s — koszt kursu dla projektantów systemów,

T_s — procent rocznego ubytku projektantów systemów.

W powyższej formule występują 3 elementy poważnie rzutujące na ogólny koszt szkolenia. Są to: koszt nauki programowania, koszt kursu dla programistów prowadzących oraz projektantów systemów.
A oto jak kształtują się programy i opłaty za przeszkolenie na kursach oferowanych przez IBM dla systemu 360.

Kurs podstaw programowania		Opłata $
podstawy systemu		30
wprowadzenie do IBM 360		30
podstawy programowania		30
system operacyjny		180
COBOL — nauka		30
COBOL — praktyka		100
	razem	400
Kurs dla programistów prowadzących		
projektowanie i analiza systemów		360
system kontroli programistów		330
kodowanie danych		400
	razem	1090
Kurs dla projektantów systemów		
kodowanie zaawansowane		460
praktyka z systemem operacyjnym		1500
	razem	1900

Wnioski

Przewiduje się, że rezultatem pogłębiającej się specjalizacji i rozproszenia wysiłków będzie wzrost kosztów szkolenia. W koncernie American Telephone and Telegraph Company ocenia się go na 7—12%, a jeden ze specjalistów Booth Computer Co., Paul William Jr., określił obecna sytuację jako największy wzrost kosztów szkolenia w historii przetwarzania danych; przewiduje on, że w przyszłości ceny urządzeń spadną o 3% przy jednoczesnym wzroście kosztów ich utrzymania o 15—20%. W roku 1951, tj. w momencie narodzin ETO, przewidywano, że wystarczy sześć głównych ośrodków obliczeniowych do realizacji światowych potrzeb w zakresie APD.
Jednakże szybki wzrost ilości urządzeń doprowadził do poważnych dysproporcji i obecnie stoimy przed faktem nienadążania w przygotowaniu ludzi do utrzymania wyprodukowanych urządzeń w ruchu.
W tym stanie rzeczy konieczna jest korekta metod szkoleniowych. Niektóre metody zostaną zachowane. Będzie wzrastać ilość kursów na terenie użytkowników komputerów z jednoczesnym wzrostem ich poziomu.
Mimo to stan pożądanego nasycenia wykwalifikowanymi programistami nigdy, niestety, nie zostanie osiągnięty.

opracował
Stefan Kwiatek
ZOWAR — Warszawa

ERRATA

Przepraszamy Czytelników i Autorów za pomyłki w zeszycie nr 9/70 naszego czasopisma:
W artykule J. Małuszyćskiego pt. „Opis składni języków programowania" — na str. 8, wiersz 9 od dołu — wzór powinien mieć następującą postać $\alpha A \beta \rightarrow \alpha \omega \beta$

W artykule S. Kobleszkowa „Pewien program do rozwiązywania problemów brydżowych" na rysunku 1 na str. 15 symbole **karo** i **kiery** powinny być zamienione między sobą, a podpis powinien brzmieć : S N grają siedem **karo** (wielki szlem).

STANY ZJEDNOCZONE AP 1970

Miasto Phoenix w Arizonie — to miasto w stanie pełnej ekspansji. 40 000 mieszkańców przed wojną, 200 000 po wojnie, 950 000 obecnie i 1 500 000 przewidzianych w roku 1980. Miasto to jest jedną ze światowych stolic elektroniki i informatyki. Wiele dokonywanych tu prac jest „ściśle tajnych", w szczególności prace związane z zamówieniami Pentagonu. Tu właśnie wyprodukowano dwa wielkie komputery GE 635, które zużytkowała NASA dla Apollo 12. Komputery te weryfikowały sprawność działania wszystkich pojazdów kosmicznych Apollo i rakiet Satrun. Te same maszyny cyfrowe mogą jednocześnie obliczać szybkość wiatru i płace pracowników ośrodka kosmicznego w Cap Kennedy.

W wielkiej betonowej klimatyzowanej hali Information System Division koncernu GENERAL ELECTRIC (czwartej co do wielkości firmy w USA, produkującej 200 000 różnych wyrobów, której obroty wynoszą 8 miliardów dolarów rocznie) znajduje sie najpotężniejszy chyba ośrodek obliczeniowy na świecie: sześć wielkich „600", wraz z urządzeniami peryferyjnymi i gigantycznymi pamięciami taśmowymi i dyskowymi. Nie opodal — fantastycznie wyposażone laboratoria kontrolne, w których inne komputery kontrolują te potężne „sześćsetki".

Ośrodek pracuje nie tylko dla 170 fabryk koncernu General Electric, znajdujących się w różnych punktach USA, lecz jednocześnie dla wielu innych klientów, rozproszonych po całym terytorium Ameryki. Ośrodek ten to wyraz nowej rewolucji informatycznej. Teleinformatyka, ti. dialog komputerów za pomocą łączy kablowych czy też satelitów ponad granicami państw i poprzez oceany — to sprawa niedalekiej przyszłości. GENERAL ELECTRIC jest drugim co do zasięgu użytkownikiem komputerów na świecie (pierwszym jest rząd amerykański). W ramach tego koncernu pracuje 350 systemów z 800 urządzeniami końcowymi, pracującymi w systemie z podziałem czasu, co oznacza 120 milionów dolarów kosztów operacyjnych rocznie. To właśnie koncern GENERAL ELECTRIC wykorzystał po raz pierwszy w jednym ze swych przedsiębiorstw w 1954 r. uniwersytecką maszynę cyfrową UNIVAC I.

Wokół ekipy pionierów UNIVAC I powstała wkrótce cała grupa informatyczna. Historia zastosowania komputerów w omawianym koncernie zamyka się w czterech wielkich datach:

- 1956 — ERMA, pierwszy komputer koncernu, skonstruowany na zamówienie i dla potrzeb Bank of America (największego banku świata),
- 1961 — magnetyczny system odczytywania czeków MICR i GE 200, pierwsza seria ekstrapolowana systemu ERMA,
- 1964 — GE 600 — maszyna wieloprogramowa,
- 1968 — 3 D, tj. trójwymiarowe wykorzystanie komputera z oprogramowaniem (GECOS III).

Ta trójwymiarowa informatyka oznacza możliwość wykorzystania wielkiego komputera trzema równoczesnymi sposobami przetwarzania informacji: klasyczne przetwarzanie lokalne, przetwarzanie na odległość za pomocą komputerów satelitarnych oraz wykorzystanie przez szereg użytkowników w tym samym czasie (time sharing). Wszystkie informacje, pochodzące z wielu źródeł, centralizowane są w jednej pamięci — w centralnej scalonej kartotece, bezpośredni zaś dostęp do danych uzyskuje się drogą telefoniczną, dzięki niewielkim urządzeniom końcowym. Aby system ten mógł sprawnie działać, konieczne jest zastosowanie właściwych metod przekazywania informacji. W tym celu pracuje się obecnie nad nowym projektem, który nosi nazwę *networking concept*, tj. uruchomienie specjalnej sieci.

W roku 1965, gdy rozpoczęto stosowanie systemów z podziałem czasu, operacje z tego zakresu wyrażały się sumą 10 milionów dolarów. W roku obecnym suma ta wynosi 150 milionów, zaś perspektywa 1975 r. wyraża się sumą około 2 miliardów dolarów. Stosowanie jednak tej metody ograniczone jest w swym zasięgu wysokością opłat telefonicznych do promienia mniej więcej 200 mil. Logicznym tego przedłużeniem jest specjalna sieć łączności. Jednostka centralna znajduje sie w Ohio, w stanie Cleveland. Została tam zainstalowana w sierpniu 1969 r. Obecnie już 40 miast amerykańskich, na wschodzie i na zachodzie, połączonych jest siecią specjalna. Można prowadzić dialog z komputerem z odległości wielu tysięcy kilometrów. 4 ośrodki będą mogły obsłużyć całe terytorium USA. Zostaną one zainstalowane — oprócz w Cleveland — w Los Angeles, w Schenectady i w Nowym Jorku. Tak np. firma GENERAL MOTORS w Detroit będzie mogła dzięki temu systemowi porozumiewać się za pośrednictwem komputerów ze wszystkimi swymi fabrykami.

Bezpośrednia komunikacja za pomocą satelitów z Europą z miasta Betshede została już urzeczywistniona. Londyn wytypowano do przeprowadzenia eksperymentu. Nie stwarza to żadnych problemów w dziedzinie technicznej, lecz koszty tego przedsięwzięcia są jeszcze ogromne. Z chwilą jednak, gdy ten problem ekonomiczny zostanie przezwyciężony, stanowić to będzie potężny oręż dla strategii przedsiębiorstw amerykańskich w Europie. W perspektywie wprowadzony zostanie również na naszym kontynencie system sieciowy z jednostką centralną, która mogłaby być zainstalowana np. w Paryżu.

Wkrótce w pełni zostanie również uruchomiony cały system trójwymiarowy. Ten *resource plan*, jeszcze nie ujawniony, znajduje sie w fazie elementarnej. Wkrótce wszystkie biblioteki programów zostana oddane do dyspozycji użytkowników. Ci ostatni są częstokroć znacznie bardziej zainteresowani zakupywaniem „godzin obsługiwania" niż „hardware'u". W początku roku 1971 system „resource" zostanie podłączony do „network". Można sobie łatwo wyobrazić, jak to w przyszłości bedzie wyglądało. Kilka olbrzymich, potężnych ośrodków informatyki w różnych punktach świata, nie ujawnionych szerokiej publiczności. Wszyscy zaś, za pewna, chyba niewielka opłata, beda mogli w sposób prosty, drogą telefoniczną, korzystać z gigantycznej, uniwersalnej pamięci. Sieć informatyki bedzie wówczas uzupełnieniem sieci telefonicznej.

Oto kilka przykładów zastosowania w USA systemu 3 D. W przedsiębiorstwach leśnictwa, obejmujących trzy i pół miliona hektarów lasu i 120 fabryk, komputer śledzi kloc drzewny od chwili jego ścięcia; w tym samym czasie opracowuje plan ponownego zalesienia. U CHRYSLERA komputer odpowiedzialny jest za mnóstwo czynności, począwszy od dywidend akcjonariuszy aż do wytłaczania karoserii. W olbrzymiej instytucji opracowującej program kosmiczny, Hughes Aircraft, 16 000 naukowców i techników pracuje wspólnie na podstawie tego samego banku danych. Federalny Departament Finansów (odpowiednik naszego ministerstwa) wykorzystuje trójwymiarowy komputer dla całej swej buchalterii podatkowej; system ten funkcjonuje ze ścisłą dokładnością. W firmie J. C. PENNEY — sieci wielkich sklepów w Kalifornii — wiele czynności administracyjnych dokonuje się błyskawicznie: komputery sprawdzają możliwości kredytowe klientów, obliczają podatki, decyduja o zakupie nowych produktów itp.

Według zdania dyrektora do spraw informatyki w firmie PHOENIX VIC CASEBOLT, w najbliższych

latach dominować będzie tendencja do maksymalnego łączenia komputerów między sobą. Wszystkie dane zostaną scentralizowane w wielkich jednostkach, z którymi zostaną połączone wszystkie urządzenia wejścia i wyjścia informacji; łączność z użytkownikami utrzymywać się będzie poprzez linie telefoniczne. Przewiduje się pojemność rzędu wielu miliardów znaków. Przed pięciu laty była mowa o milionach. Odpowiednio zmniejszą się koszty przechowywania informacji. Obecnie najszerzej stosowane są komputery średniej wielkości. Wkrótce użytkowane będą przede wszystkim najmniejsze i największe.

Ostatni, rzucony niedawno na rynek, komputer GE 655 o wysokiej wydajności może pracować w bezpośredniej łączności z 384 stacjami końcowymi. Część centralna, sterowanie, moduły pamięci, moduły sterowania wejściami i wyjściami stanowią odrębne jednostki, pozwalające użytkownikom na przystosowanie całego systemu do swych własnych potrzeb.
Około sześćdziesięciu komputerów trójwymiarowych jest już w użyciu w USA. Stosuje się je w NASA, w Pentagonie i w czterech największych firmach amerykańskich: GENERAL MOTORS; FORD; STANDARD OIL i oczywiście GENERAL ELECTRIC. Są one już również stosowane na wyższych uczelniach, takich jak Darmouth College, który był pionierem informatyki wykorzystywanej do celów dydaktycznych, czy w Massachusetts Institute of Technology. Dwanaście komputerów tego typu znajduje się poza Stanami Zjednoczonymi AP. Przewiduje się, że w 1975 r. funkcjonować będzie na całym świecie przeszło 1500 takich komputerów.

Źródło: art. J. Creisera — Etats — Unis 70: La nouvelle revolution de l'ordinateur, Le Figaro, 2.I.70

K. P.

KONFERENCJE, WYSTAWY, TARGI

Rok 1970

7—12.XII. Węgry — Budapeszt. Spotkanie robocze na temat: „Budowa systemów operacyjnych maszyn cyfrowych" Centrum Obliczeniowe Węgierskiej Akademii Nauk

Rok 1971

29.III.—2.IV. Francja — Paryż. Colloque International: "L'Espace et la Communication" — collecte et transmission de l'information dans les systemes spatiaux d'application". Secretariat: 16, rue de Presles, 75-Paris 15-, France.

29.III—2.IV. Anglia — Nottingham. British Computer Society Data Fair. 23 Dorset Square, London, N. W. 1, England.

I kwartał — Polska — Warszawa. Sympozjum naukowe pt.: „O metodach przetwarzania informacji niesfomalizowanej" (2 dni). Zarząd Główny SEP, Warszawa, ul. Czackiego 3/5.

25.—30.X. Jugosławia — Zagrzeb. Internat. Exhibition of Office Equipment — Interbüro.

H. Kl.

PRETO informuje...

EUROPEJSKI PROGRAM BADAWCZY DIEBOLDA

Od r. 1967 Biuro PRETO uczestniczy w Europejskim Programie Badawczym Diebold *(Diebold Research Program Europe — DRPE)*. DRPE jest formą działalności filii amerykańskiego przedsiębiorstwa doradców w dziedzinie organizacji i zastosowań ETO. Program Diebolda stanowi forum wymiany doświadczeń między wielkimi użytkownikami EMC. Udział w nim biorą największe koncerny światowe *(Dupont, Standard Oil, Nestle, Shell, British Aircraft, Philips, IBM, Siemens itp.)* oraz instytucje państwowe (Electricite de France, General Post Office UK itp.). Z krajów socjalistycznych w programie uczestniczą CSRS (Tesla) i NRD *(VVB Datenverarbeitung und Büromaschinen)* zaś Bułgaria jest w trakcie zawierania umowy.

W wyniku uczestnictwa w DRPE Biuro PRETO otrzymuje materiały opracowywane w ramach programu (dotychczas otrzymało około 70 pozycji z czego przetłumaczone zostało 10 a dalsze 10 znajduje się w tłumaczeniu) oraz uczestniczy w konferencjach i spotkaniach organizowanych przez DRPE (cztery konferencje i jedno spotkanie na szczeblu kierowniczym rocznie). Materiały przed i pokonferencyjne obejmują około 50 pozycji wydawniczych. Wszystkie powyższe materiały znajdują się w Bibliotece PRETO. Dotyczą one zagadnień związanych ze stosowaniem ETO począwszy od środków technicznych do epd, poprzez technologię pd, metodologię projektowania systemów epd, teorię zarządzania w aspekcie powiązań z elektronicznym przetwarzaniem danych, szkolenie kadr kończąc na szerokim stosowaniu ETO np. w dydaktyce, medycynie itp.

Zagadnieniem prawidłowego wykorzystania materiałów i informacji uzyskiwanych w ramach przynależności do DRPE zajmuje się powołana przez Pełnomocnika Rządu do Spraw ETO rekrutujący się z różnych instytucji w Polsce. Uzyskiwane materiały wykorzystywane są przy opracowywaniu typowych systemów w Biurze Studiów i Projektów SEPD, w pracach nad programami rozwoju informatyki w branżach, w projektowaniu systemów EPD względnie w ocenie celowości realizacji gotowych projektów.

Z ciekawszych pozycji przetłumaczonych na język polski można wymienić informacje o zintegrowanym systemie informacyjnym do celów zarządzania — IMIS, badania operacyjne a kierownictwo, testy i selekcja personelu. Do ciekawszych konferencji, w których uczestniczyli polscy specjaliści, można zaliczyć konferencję w Rzymie w dniach 2—7 marca 1969 r., na której m. in. wymieniano poglądy na zagadnienia automatycznego przetwarzania danych; bieżący stan wdrożenia systemu „SID" w firmie Dalminie (Mediolan) oraz wymagania użytkowników komputerów czwartej generacji.

Sprawozdania z uczestnictwa przedstawicieli Polski w tych konferencjach znajdują się w bibliotece PRETO.

KONFERENCJA W HANOWERZE

W dniach 9—11 czerwca 1970 odbyła się w Hanowerze, NRF, XIX Konferencja Europejskiego Programu Badawczego Diebolda, w dwóch sesjach.

Sesja I zajmowała się problematyką wyboru systemu.

Sesja II omówiła zagadnienia związane z zarządzaniem i nadzorem projektowanych systemów EPD.

Zrówno w sesji I jak i II, referenci omówili szereg praktycznych problemów oraz podali własne wnioski.

Na przykład podczas sesji dotyczącej wyboru systemu z naciskiem podkreślano konieczność odpowiedniego uzasadnienia podjęcia nowego zastosowa-

nia komputerów (dlaczego chcemy opracować system) oraz określenie we wstępnym opracowaniu, jak będzie to zastosowanie wykorzystane (co uzyskamy przez jego wdrożenie) i jaki jest cel opracowania systemu.

W sesji zajmującej się problematyką zarządzania i nadzoru nad projektowaniem systemów elektronicznego przetwarzania danych omówiono na podstawie istniejących doświadczeń struktury organizacyjne komórek projektowych oraz zagadnienia związane z projektowaniem i wdrażaniem systemów. Przeważała opinia, że sprawna realizacja prac projektowych i wdrożeniowych jest możliwa jedynie wówczas, gdy nad całością prac czuwa naczelne kierownictwo instytucji, w której ma być wdrożony system elektronicznego przetwarzania danych.

Na zakończenie konferencji odbyło się seminarium p.t. „Która z metod wyboru systemu jest najkorzystniejsza". Na sympozjum przeważał pogląd na temat metod wyboru systemu, popartego analizą potrzeb, i że metoda komparatystyczna najlepiej się nadaje do systemów w małych zakładach i instytucajch, zaś symulacyjna do wyboru dużych systemów przetwarzających na bieżąco. Z podanych na sympozjum danych statystycznych wynikało jednak, że znakomita większość przedsiębiorstw i instytucji np. we Francji opierała się na rozeznaniu praktycznym rezygnując, albo nie wykorzystując wyników otrzymanych za pomocą powyższych metod wyboru.

W drugim dniu konferencji kanclerz NRF wygłosił exposé powitalne. Organizatorzy konferencji umożliwili także uczestnikom kontakt telefoniczny (połączenie i rozmowę) z dyrektorem Centrum Marshalla z NASA, Stany Zjednoczone (program APOLLO).

Następna XX Konferencja Europejskiego Programu Badawczego Dibolda odbędzie się pod koniec IV kwartału br. na tematy:

a) mieszane konfiguracje hardwarowe

b) zastosowanie komputerów w planowaniu i kontroli finansowej

SYLWETKA I ZAKRES NAUCZANIA PROJEKTANTA SEPD

Z inicjatywy Biura Pełnomocnika Rządu do Spraw Elektronicznej Techniki Obliczeniowej w porozumieniu i przy współudziale Ministerstwa Oświaty i Szkolnictwa Wyższego oraz Komisji Organizacji i Zarządzania przygotowuje się na połowę grudnia br. seminarium poświęcone sylwetce i zakresowi nauczania projektantów (organizatorów) systemów elektronicznego przetwarzania danych. Aktualnie wiele kłopotu sprawia brak definicji profilu zawodowego w zakresie projektowania systemów elektronicznego przetwarzania danych. Seminarium odbędzie się w Centralnym Ośrodku Doskonalenia Kadr Kierowniczych. W przygotowaniu jego biorą udział wybitni polscy specjaliści w tej dziedzinie, zarówno pracownicy nauki, jak również praktycy zatrudnieniu w ośrodkach obliczeniowych.

Do opracowania tematu: SYLWETKA PROJEKTANTA (ORGANIZATORA) SYSTEMÓW ELEKTRONICZNEGO PRZETWARZANIA DANYCH (INFORMACJI) zostali zaproszeni:

- mgr inż. Zdzisław Bogdanowicz
  — dyrektor ZETO Szczecin
- mgr inż. Mieczysław Brewka
  — szef Zakładowego Ośrodka Obliczeniowego Stoczni im. Lenina w Gdańsku
- mgr inż. Jerzy Żydowo
  — dyrektor ZETO Gdynia

Na koreferentów tego tematu zaproszono:

- dr Zb. Gackowskiego
  — z-cę dyrektora d.s. Naukowych Instytutu Organizacji Przemysłu Maszynowego w Warszawie
- dr B. Warzecha
  — dyrektora d.s. Ekonomicznych H.P.M.O.A. — w Katowicach

Do tematu: ZAKRES NAUCZANIA PROJEKTANTÓW (ORGANIZATORÓW) SYSTEMÓW ELEKTRONICZNEGO PRZETWARZANIA DANYCH (INFORMACJI) zostali zaproszeni:

Referat I

mgr Jerzy Trybulski
Wyższa Szkoła Ekonomiczna we Wrocławiu

Koreferat:

prof. dr hab. Tadeusz Peche kierownik Katedry Przetwarzania Danych SGPiS

Referat II

mgr inż. Zb. Drabek — kierownik Działu Szkolenia Centralnego Ośrodka Doskonalenia Kadr Kierowniczych

Koreferat:

dr Z. Zapolski
Biuro Studiów i Projektów Systemów Elektronicznego Przetwarzania Danych — ZETO.

Celem seminarium jest przygotowanie materiałów umożliwiających Ministerstwu Oświaty i Szkolnictwa Wyższego utworzenie nowego profilu zawodowego, w którym będą kształcone kadry informatyków. Ponadto chodzi również o określenie i przedyskutowanie zakresu oraz treści nauczania (szkolenia) kadr w tym profilu zawodowym.

W seminarium udział wezmą przedstawiciele szkół wyższych, ośrodków obliczeniowych oraz zainteresowanych resortów.

SZKOLENIE LICENCYJNE W WZE „ELWRO"

Wraz z uruchomieniem seryjnej produkcji EMC ODRA-1304 Wrocławskie Zakłady Elektroniczne „ELWRO" przystępują do szkolenia kadr użytkowników tego komputera. Wprowadza się również pewną innowację polegającą na tym, że koszt szkolenia 24-osobowej ekipy włącza się do ceny sprzedażnej.

Dotychczasowa praktyka polegająca na tym, że WZE „ELWRO" udzielało rabatu na szkolenie personelu, napotykała na bardzo poważne przeszkody w jej realizacji. Przedsiębiorstwa nie posiadają z reguły limitów szkoleniowych przewidzianych na opłacenie uczestnictwa w wielomiesięcznym szkoleniu swoich pracowników. Wprowadzona zmiana wychodzi naprzeciw użytkownikom i dotyczy również maszyn do obliczeń numerycznych . ODRA-1204, na którą producent zobowiązany jest do szkolenia 8-osobowej ekipy.

W sprawach szkolenia licencyjnego kadr dla komputerów produkowanych przez WZE „ELWRO" należy zwracać się pod adresem: „ELWRO — SERWIS" — WROCŁAW ul. Ostrowskiego 30.

Józef Śnieciński
Biuro PRETO

CZYTELNIKU!

Czy wznowiłeś prenumeratę miesięcznika

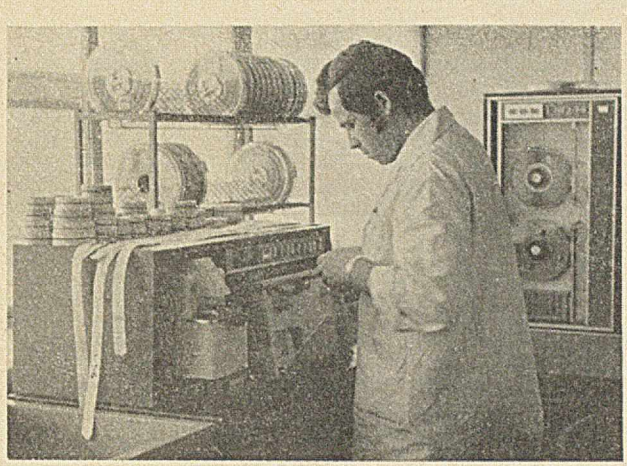
„MASZYNY MATEMATYCZNE"

na rok 1971 ?

SEMINARIA KLUBU UŻYTKOWNIKÓW KOMPUTERÓW RODZINY ICL-1900

Już od dwóch lat działa formalnie akredytowany przy Polskim Komitecie Automatycznego Przetwarzania Informacji stowarzyszenie użytkowników komputerów rodziny ICL-1900. Stowarzyszenie to, zwane potocznie **„Klubem ICL-1900"**, a którego sekretariat prowadzi Zakład Elektronicznej Techniki Obliczeniowej ZETO w Gdyni, odbywa okresowe spotkania seminaryjne, organizowane — z założenia — kolejno przez wszystkie ośrodki członkowskie.

Rzeczą, która niewątpliwie uderza postronnego obserwatora jest brak, w zasadzie, aktu prawnego precyzującego zakres i formy współpracy między ośrodkami użytkującymi komputery rodziny ICL-1900. Współpraca ta odbywa się bowiem na zasadach samopomocy, przewidzianej zresztą w statucie Klubu. Jak dotychczas, samopomoc ta zdała egzamin. Przedmiotem pomocy były zarówno części zamienne, które jedne ośrodki wypożyczały drugim (do czasu zrealizowania dostaw uzupełniających), jak i uruchomione programy biblioteczne.



I Seminarium

Jednym z ważniejszych elementów tej samopomocy są właśnie wspomniane seminaria, z których ostatnie odbyło się w Ośrodku Elektronicznym GUS w Warszawie na temat organizacji i planowania procesu przetwarzania danych. Poprzednie zaś i pierwsze w ogóle seminarium odbyło się w Gdyni w dniach 5 i 6 lutego 1970.

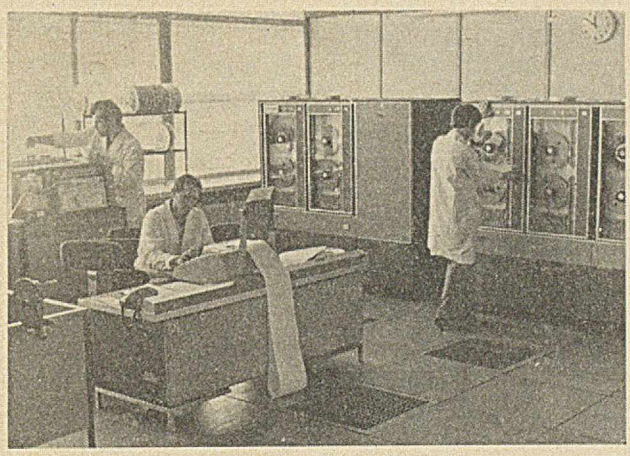
Właśnie temu pierwszemu seminarium chcemy tutaj poświęcić nieco więcej uwagi wobec dosyć bogatej tematyki objętej jego programem. W seminarium tym wzięło udział około 50 osób reprezentujących sześciu ówczesnych krajowych użytkowników komputerów ICL-1900, trzech użytkowników in spe — oraz kilku obserwatorów z ramienia instytucji centralnych. Samo seminarium rozpoczęło się od zapoznanania uczestniczących w nim osób z ośrodkiem ZETO-Gdynia.

Mgr inż. T. Mazurkiewicz w blisko dwugodzinnym referacie omówił wyczerpująco działalność Ośrodka, skupiając uwagę na organizacji pracy, ze szczególnym podkreśleniem zadań SEKCJI KONTROLI; podał też przykładowo dane liczbowe charakteryzujące pracochłonność wykonywanych obliczeń. I tak np. w wyniku wieloprogramowego wykorzystania EMC 600 godzinom pracy (fizycznym) maszyny w styczniu br. odpowiadało około 880 godzin pracy jednoprogramowej (przeliczeniowo). Prowadzenie pełnej ewidencji materiałowej dla 12 gdańskich przedsiębiorstw budowlanych (wraz z elementami rozliczania poszczególnych budów) absorbuje przy tym co miesiąc około 42 maszynogodzin (przeliczeniowych), z tym, że część nośników przygotowywana jest w ZETO, a część bezpośrednio przez klientów. Prelegent omówił też w skrócie dalsze plany rozwojowe ZETO-Gdynia. Będą one — rozwijać się w dwóch kierunkach: 1) rozbudowa istniejącego ośrodka (dobudowanie nowego skrzydła do istniejącego budynku adaptowanego z lokali biur handlowych) oraz 2) budowa nowego trójkomputerowego ośrodka w Gdańsku przy ul. Heweliusza. Liczne szczegóły o ZETO-Gdynia znane są zresztą Czytelnikom naszego czasopisma [1].

Dla odprężenia przed następnym referatem, uczestnicy mieli możność obejrzeć film o możliwościach zastosowania ekranopisów przy projektowaniu budynków mieszkalnych z elementów prefabrykowanych. Film ten (wypożyczony z Biblioteki COPAN w Warszawie) niezbyt, co prawda, wiązał się z tematyką seminarium, niewątpliwie jednak w znakomity wizualnie sposób oddawał istotę projektowania na „elektronicznej rysownicy" (szkoda tylko, że Biblioteka COPAN nie dorobiła się jeszcze innego, niż prowizoryczne tłumaczenia objaśnień angielskich, odtwarzanego z magnetofonu).

Referat mgr inż. J. Malickiego — **„Technika projektowania a strategia wdrażania systemów EPD"** — stanowił świetny przykład, w jaki sposób nudne wyliczanie stu kilkudziesięciu elementów projektu systemowego można zastąpić efektownymi ruchomymi napisami. Prelegent po prostu — zamiast drogiego, importowanego flanelografu, jakim w Polsce dysponuje chyba tylko telewizja (używając go w dodatku niezwykle rzadko w obawie przed zbyt szybkim zniszczeniem) — użył zwykłej zielonej flaneli i plansz podklejonych papierem ściernym. I w ten sposób, unikając niewyraźnego pisania kredą na sfatygowa-

(dok. na IV str. okł.)



[1] P. „Maszyny Matematyczne" nr 7—8/69 — zeszyt poświęcony problemom ETO w gospodarce morskiej, przygotowany przy współpracy ZETO-Gdynia.

(dokończenie z III str. okł.)

nej tablicy — omówił cały cykl czynności i etapów projektowania — od wizyty kierownictwa instytucji zamierzającej być klientem ZETO aż do sporządzenia katalogu dokumentacji w 6 egzemplarzach (Czytelnicy darują, że nie wypiszemy wszystkich elementów, gdyż z jednej strony zabrakłoby na to miejsca, a z drugiej — prelegent przygotowuje osobny artykuł na ten temat).

Bardzo konkretny i przydatny w pracy zaawansowanych użytkowników referat na temat **wykorzystania standardowych pakietów** programowych wygłosił mgr J. Rucki. Niestety referat nie był uzupełniony streszczeniem, a zawierał mnóstwo informacji szczegółowych, zaczerpniętych z nie wydanego jeszcze wówczas tłumaczenia dokumentacji firmowej (pakiety STATISTICAL ANALYSIS, SIMPLEX, TRANSCH, DEPOSITING i inne). Prelegent podał szereg liczb charakteryzujących czasochłonność rozwiązywania różnych zadań przykładowych, jakie badano w Gdyni podczas uruchamiania odnośnych pakietów.

W drugim dniu seminarium wygłoszono siedem referatów, omawiających wdrożone w ZETO-Gdynia oprogramowanie firmowe:

- **Automatyczna kompilacja i korekta programów**
- **Kompilatory seryjne języka PLAN**
- **Kompilatory seryjne języka COBOL**
- **Kompilatory seryjne języków ALGOL i FORTRAN**
- **System PATSY automatycznego testowania**
- **System BATCH MONITOR dla języków ALGOL i FORTRAN**
- **Automatyczne testowanie w systemie GEORGE**

(mgr inż. R. Jeliński, mgr A. Czerwiński, mgr inż. A. Szymborski).

Przygotowane zawczasu obszerne streszczenia tych referatów ułatwiły zainteresowanym zrozumienie (i zapamiętanie!) poruszonych kwestii.

Na zakończenie seminarium odbyła się generalna dyskusja, w której poruszono szereg spraw dość istotnych dla członków Klubu ICL-1900, aczkolwiek znacznie przekraczających formalnie poza ramy roboczego spotkania, m.in. zwrócono uwagę, że:

- przyszli użytkownicy komputerów jeszcze nie wiedzą, jakie maszyny otrzymają w najbliższych latach
- ewidencja czasu pracy i rozliczeń kosztów, przewidziana Zarządzeniem nr 44 Przewodniczącego KNiT właściwie uniemożliwia prowadzenie tego zagadnienia na komputerach
- dotychczas nie ujednolicono terminologii projektowania systemów EPD
- obowiązujący taryfikator praktycznie uniemożliwia zatrudnianie w ośrodkach obliczeniowych „branżystów" dotychczas nie pracujących w ETO.

Niewątpliwie sprawy te powinny być szeroko i szczegółowo zbadane na forum PKAPI.

ABE

XVIII Międzynarodowy Kongres Naukowy Elektroniki

W Rzymie od 29 do 31 marca 1971 r. odbędzie się XVIII Międzynarodowy Kongres Naukowy Elektroniki pod hasłem: **„Stan obecny i postęp w elektronice".**

W zaproszeniu do zgłaszania referatów organizatorzy zaznaczają, że nie stawia się żadnych ograniczeń tematycznych, a Komitet Techniczno-Naukowy Kongresu będzie kwalifikować referaty według kryteriów oryginalności (z uwzględnieniem możliwości organizacyjnych). Ze swej strony Komitet proponuje następującą tematykę, którą uważa **za najbardziej interesującą** w chwili obecnej:

A. **Wielkie systemy łączności pracujące przy częstotliwościach powyżej 10 GHz**

B. **Nowoczesne urządzenia pamięciowe**

Tematyka obejmuje zarówno najnowszy rozwój pamięci tradycyjnych (taśmy, bębny i dyski magnetyczne; operacyjne pamięci rdzeniowe, cienkowarstwowe i półprzewodnikowe), jak też nowe typy pamięci (optyczne, holograficzne i kriogeniczne) oraz problemy stosowania tych urządzeń w komputerach i elektronicznych centralach telefonicznych.

C. **Urządzenia audiowizualne i komputerowe do celów nauczania**

— zastosowanie do nauczania teoretycznego i kursowego (szkoły, zakłady pracy itd.), z uwzględnieniem aspektów technicznych i ekonomicznych problemu oraz wyników metodologicznych.

D. **Najnowsze osiągnięcia techniki transmisji danych**

— ze szczególnym uwzględnieniem działania i techniki wykonania pracujących lub przewidywanych na przyszłość różnych typów urządzeń końcowych dla ośrodków przetwarzania danych.

E. **Przesyłanie telewizyjnych sygnałów PCM**

Termin zgłaszania referatów (tytuł wraz ze streszczeniem) — do 1 grudnia 1970 r.

Termin nadsyłania tekstów referatów, po ich przyjęciu przez Komitet Naukowo-Techniczny Kongresu — do 10 lutego 1971 r.

Adres Sekretariatu Kongresu: 00193 Roma — Via Crescenzio, 9, ITALIA.